Robust Networks
Modeling & Scheduling

by

Shaikh Arifusalam

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEMS ENGINEERING

December, 1998

ROBUST NETWORKS MODELING & SCHEDULING

BY

SHAIKH ARIFUSALAM

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEMS ENGINEERING

DECEMBER 1998

UMI Number: 1398018

UMI®

UMI Microform 1398018

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by

SHAIKH ARIFUSALAM

under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

Thesis Committee

Dr. Shokri Z. Selim (Chairman)

Dr. Salih O. Duffuaa (Member)

2/6/99

Dr. Omar M. Al – Turki (Member)

11-6-99

Dr. Kamal Al – Harbi (Member)

14/6/99

Dr. Abdulbasit Andijani
Department Chairman

Dean of Graduate Studies

16/6/99
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedicated

to

my

Parents, Sisters

&

Grand Father

for their patience and support.

Acknowledgements

Praise be to Allah(swt), the lord of the World, the Almighty for having guided me at every stage of my life.

Words are not adequate to describe my gratitude towards my advisor Dr. Shokri Z. Selim. It was a great pleasure working with him. I would also like to place on record my appreciation for the cooperation and guidance extended by my committee members, Dr. Salih O. Duffuaa, Dr. Omar Al-Turki, and Dr. Kamal Al-Harbi.

I also acknowledge the generous help and support for this research given by KFUPM. My thanks to the Chairman and faculty members of my department for their support. I would like to thank Mr. Moizuddin for his kind help. My thanks to my friend and room mate Abul Bashar, my house mates Farhan and Shafique. My special thanks to Waseem, Akram, Aijaz, Shoukat, Razzack, Adeeb, Samad, Haleem, Mukarram, Shahed, Shareef, Nazeer, Azmath and all others whose encouragement and support throughout my stay in KFUPM made it a pleasant and memorable one.

Contents

List of Tables

List of Figures

List of Algorithms

THESIS ABSTRACT

Name: SHAIKH ARIFUSALAM

Title: ROBUST NETWORKS MODELING & SCHEDULING

Degree: MASTER OF SCIENCE

Major Field: SYSTEMS ENGINEERING

Date of Degree: DECEMBER 1998

It is usually assumed that the parameters of an optimization problem to be known numbers or scalars. Accordingly, the objective function value is a scalar or a point value. However, in real life, the parameters may not be known exactly and all we know about them is that they fall in a given interval. We consider the activity/job duration times as interval values and solve several network and scheduling problems. Under network problems we show results for critical path method (CPM), crashing network activity, minimal spanning tree problem, transportation and assignment problems. Under scheduling problems we have considered minimizing the mean flow time problem, minimizing the maximum lateness problem, minimizing the number of tardy jobs problem, minimizing the sum of earliness and tardiness of all jobs under a given schedule problem and minimizing the weighted sum of earliness and tardiness for jobs with different weights for a given schedule.

Keywords: *interval, network problems, scheduling problems.*

King Fahd University of Petroleum and Minerals, Dhahran.
DECEMBER 1998

خلاصة الرسالة

الاسم : شيخ عارف السلام

العنوان : تصميم وجدولة الشبكات المكينة

الدرجة : ماجستير في العلوم

التخصص : هندسة النظم

تاريخ الشهادة : ديسمبر ١٩٩٨م.

من المألوف في مسائل الأمثية أن يفترض لمعانها أخذ قيم عددية أو قياسية معلومة. وبالتالي فإن دالة الهدف تأخذ قيمة عددية أونقطية. على أنه في التطبيقات الحياتية يمكن أن تكون هذه المعالم (المتغيرات) غير معروفة بالضبط بل كل ما يعرف عنها أنها تقع في حدود فترة معطاة في هذه الرسالة قمنا باعتبار زمن كل نشاط عنى أنه فترة وحللنا العديد من مسائل الشبكات والجدولة. بالنسبة لمسائل الشبكات أعطينا نتائج لطريقة المسار الحرج (CPM) ، مسائل الاختزان، مسألة تقليل التأخر الأعظم، مسألة تقليل عدد الأعمال المتأخرة، مسألة تقليل مجموع تبكير وتأخير كل الأعمال تحت جدولة معطاة وتقليل المجموع الموزون لتبكير وتأخير الأعمال بأوزان مختلفة لجدولة معطاة.

ماجستير في العلوم

جامعة الملك فهد للبترول والمعادن

الظهران – المملكة العربية السعودية

ديسمبر ١٩٩٨ م

Chapter 1

Introduction

1.1 Historical Background

Network flow models and solution techniques provide a rich and powerful framework from which many engineering problems can be formulated and solved. Network analysis and techniques can be of great value in the design, improvement, and rationalization of complete large scale systems. The origins of network analysis are old and diverse dating to 1736 when Leonard Euler formulated the solution of the famous Konigsberg bridge [14]. More than a century later, James Clerk Maxwell and Qutsar Robert Kirchoff discovered certain basic principles of network analysis in the cause of their studies of electric circuits. Pioneering work in the modern network analysis was conducted by Hitchcock [22] in 1941 and Koopmans [32] in 1947. Since the development of the simplex algorithm by George Dantzig in 1947, network analysis has been extensively studied. The emphasis on research in 1950s and 1960s was on formulation of new models and development of new algorithms.

Network models and analysis are widely used in operations research for diverse applications such as production-distribution systems, military logistics systems, urban traffic systems, railway systems, communication systems, pipe network systems, facilities location systems, file merge systems, routing and scheduling systems, electrical networks, etc. [24].

1.2 Project Management

A project is defined as a combination of interrelated activities that must be executed in a particular order (precedence relationships) to complete an entire task. One of the popular ways of representing precedence relationships of a project is through the use of networks. In this "project network" the arcs can be used to represent the individual activities and the nodes represent the events which denote the completion time of project activities.

Project management is concerned with the scheduling and control of activities in such a way that the project can be completed as soon as possible after its start. Project planning using the activity network representation is called critical path method (CPM).

The total cost of the project is the sum of the direct costs (inversely proportional to the activity times) and the indirect costs (proportional to the project duration). The CPM essentially studies the trade-off between the total cost of the project and its completion time. The project management problem is to determine the amount by which the various jobs are to be crashed which will minimize the total cost of the project (direct and indirect).

1.3 Scheduling Problems

Scheduling is a decision making process that exists in most manufacturing and production systems as well as in most information processing environments. Scheduling concerns with the allocation of limited resources to tasks over time, while achieving optimization of one or more objectives [27].

In simple terms a number of jobs is to be sequenced and time tabled in order to achieve a certain objective is called Scheduling. Each job needs to be processed on a certain number of machines in a certain order. Scheduling problems range from single machine to multiple machines with different shop structure and routing requirement. The objective is to optimize certain measure of performance such as maximum flow time, mean flow time, earliness, tardiness of jobs, etc., The basic things needed for the scheduling of any jobs are the number of jobs, their processing times and their ready times. Some problems require that the due date of each job be given so that the objectives are determined accordingly.

The assumptions and notations are elaborated in Chapter 4 where we deal with some scheduling problems for interval processing times. One can find many books and very rich literature in scheduling, however the uncertainty is not considered much. In Chapter 4 we have included the interval processing times for solving some basic single machine scheduling problems like minimizing flow time, minimizing maximum lateness, minimizing tardiness, minimizing number of tardy jobs, minimizing the sum of earliness and tardiness for all jobs, minimizing weighted sum of earliness and tardiness for jobs with different weights, etc.

1.4 Objectives of the Study

The purpose of this study is to introduce the interval activity/project duration times in the traditional network and scheduling problems and develop efficient algorithms for this approach. The Objectives of this study can be enumerated as follows :

1. To develop an algorithm which will give the critical path and the critical activities for the interval activity duration times.

2. To solve the minimal spanning tree problem for the interval approach and hence develop algorithms for the proposed objectives.

3. To develop models to solve the transportation and assignment problems for the case where costs are interval-valued.

4. To solve the crashing of networks problem by embedding the interval approach in general network crashing problems.

5. To develop algorithms for minimizing the mean flow time problem for interval processing times.

6. To develop algorithms for minimizing the maximum lateness problem for interval processing times.

7. To develop algorithms for minimizing the number of tardy jobs problem for interval processing times.

8. To develop approach for minimizing the sum of earliness and tardiness for all jobs for interval processing times for a given schedule.

9. To develop approach for minimizing weighted sum of earliness and tardiness for jobs with different weights, with interval processing times for a given schedule.

1.5 Organization of the Thesis

The organization of this thesis is as follows. The interval approach and the options of the decision maker will introduced in Chapter 2. A theorem which is valid for many problems considered in this thesis will also proved in this chapter. In Chapter 3 network problems will be studied where the activity times are given as intervals and solutions will be discussed for CPM, networks crashing problem, the minimal spanning tree problem, transportation and assignment problems. In Chapter 4 some single machine scheduling problems, namely, minimizing flow time, minimizing maximum lateness, minimizing tardiness, minimizing number of tardy jobs and the problem of minimizing the sum of earliness and tardiness for all jobs, minimizing the weighted sum of earliness and tardiness for jobs with different weights will be solved considering interval processing times for each job. Finally, the summary and conclusions will be discussed in Chapter 5.

Chapter 2

Problem Definition

2.1 Introduction

In this chapter we give the idea of how the optimization problems can be solved when their parameters are given as interval values. In Section 2.2 we explain the concept of interval approach and explain how the parameters of an optimization can be given as intervals to be more practical. In Section 2.3 we give a brief survey of literature in the field of uncertainty, while considering the activity times or the processing times. Since we are dealing with interval parameters we now have more than one objectives for different optimization problems. These objectives are discussed in Section 2.4. A theorem which is common to many problems on hand will be proved in Section 2.5. In Section 2.6 we will briefly discuss about how the solution for Objective 4 is related with the solution of Objective 2. The conditions for the selection of the desired limits to solve any problem for Objective 5 is given in Section 2.7.

2.2 The Interval Approach

Whenever we consider the parameters of a certain optimization problem we need to have estimated values for these parameters, through which we can find the objective value of the problem. However, in real life the parameters may not be known exactly and we only know that they fall in a given interval. Assuming that the parameters are given in the form of an interval, we solve different problems for different objectives and obtain solutions which also fall in an interval form.

The traditional way of looking at these estimates is obtaining a point estimate, based on which the whole schedule is developed. In real world problems we often face difficulty to stick to this point estimate because of the uncertainty in the completion time(processing time) of an activity(job). To be more practical we assume that an activity is given an interval duration estimate, i.e., a lower limit and an upper limit, for the duration of the activity. This modelling approach is now proposed to solve different network and scheduling problems.

In project management the project analysts may not be able to specify the project activity times with certainty, which is sometimes due to the inherent random nature of the activity duration. The uncertainty may also be due to a sheer lack of information about the activity, which may occur if the firm takes on an activity which it has never performed before [8]. In this case, there is only a vague idea about the activity durations, which then must be estimated as intervals. The same argument can be made for job shop scheduling problem.

2.3 Literature Survey

Researchers have always been in the search for new techniques to include uncertainty for the activity duration times in the network analysis and project management. PERT summary report (1958) was the first to be published report which gave the technique to deal with the uncertainty in project management. PERT was always criticized and objections were raised regarding their assumptions about activity duration times in the project (beta-distributed random variables), the technique for deriving parameters and also their method of determining project completion time.

Authors proposed the assumption of randomness in activity realization times. This led to the possibility to include all the more known probability distributions for activity realization times in the project (as e.g., triangular distribution [36], a poisson distribution [43], a normal distribution [33]). Many authors (Clark, 1961: Fulkerson, 1962; MacGrimmin and Ryavec, 1964; Martin, 1965: Hartley and Wortham, 1966: Lootsma, 1960, Elmaghraby, 1967, 1977; Burt and Garman, 1971) tried to improve the PERT by a more sophisticate, but nevertheless approximate maximization, and some by simulation. In 1966, Lootsma F.A. [34] gave the extended stochastic version of the PERT.

It was after the pioneering work of Zadeh, 1965 [53] in fuzzy-set theory that the research was directed towards the new approach to deal with uncertainty. It was after Zadeh's work that researchers thought whether stochastic models yield a proper representation of the uncertainty in the project. Dubois and Prade(1978a) [9], established arithmetic operations on fuzzy numbers and developed modelling with fuzzy concepts. They also applied this idea in the PERT (1978b).

Apart from the above discussed approaches to deal with uncertainty, researchers are coming up with new ideas to tackle the problem. Bintong et.al., [2] consider the uncertainty in the activity duration times and hence assume that the actual activity falls in a given interval. They model the activity durations as intervals defined by a lower and upper bound, and don't assume any probability distribution. They propose that a robust schedule can be obtained by applying a certain worst case analysis for all the possible scenarios (a set of activity durations). Thus, a robust schedule is the one which is likely to be the least deviated from the optimal schedule.

2.4 Decision Maker's Options

Depending upon the problem, parameters like duration times, cost, or distance are required to solve a problem. For the proposed approach we need the parameters in an interval form. The input is given in the form $[l_i, u_i]$, where l_i is the lower limit of activity/job and u_i is the upper limit of activity/job i. Figure 3.3 shows that when the input is given in the form of intervals, the objective is also obtained in the form of an interval $[L, U]$.

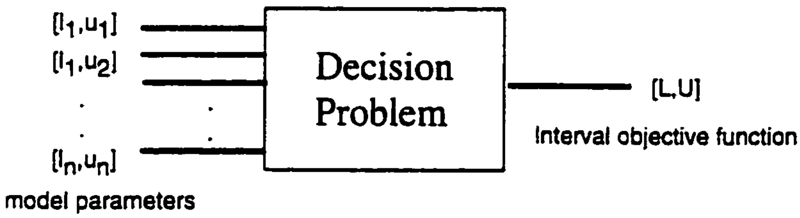

Figure 2.1: Decision Problem

The proposed approach leads to a list of possible objectives. In the traditional approach there used to be only one objective for a given problem. We propose the list of following options for the decision maker.

1. OBJECTIVE 1 :To minimize the lower limit of the objective function(L).

2. OBJECTIVE 2 :To minimize the upper limit of the objective function(U).

3. OBJECTIVE 3 :To minimize the variation or uncertainty in the objective function which is to minimize the width of the interval(U - L).

4. OBJECTIVE 4 :To minimize the lower limit of the objective function provided that the upper limit is minimized.

5. OBJECTIVE 5 :To get a solution within a desired interval $[L_d, U_d]$.

6. OBJECTIVE 6 :To get a robust schedule.

The last objective is to obtain a robust schedule. A *robust schedule* can be defined as a schedule which has the minimum deviation from the optimal schedule. This topic will be explained in detail in Section 4.2. In the following we present some network and scheduling problems where the problem parameters are in interval form.

2.5 Minimizing the width of the Objective function

In this subsection we introduce a theorem that applies to objective functions of the form. $f = \sum c_j x_j$ where c_j are interval numbers.

Theorem 2.5.1:

Consider the problem of *minimize* $f(x) = \sum_{l=1}^{m} c_j x_j$ subject to $g_i(x) \leq 0 \quad 1 \leq i \leq m$ where $c_j = [c_j^l, c_j^u]$. For a given x let $f(x) = [f^l. f^u]$. Then the minimum of $f^u - f^l$ is obtained by solving the problem

$$Minimize \ \sum_{l=1}^{m} (c_j^u - c_j^l) x_j$$

subject to

$$g_i(x) \leq 0 \quad 1 \leq i \leq m \tag{2.1}$$

Proof

Consider a feasible point $\hat{x}$, i.e. $g_i(\hat{x}) \leq 0$, then we obtain $f = [f^l(\hat{x}), f^u(\hat{x})]$. The width of the objective function is given by

$$w_f = f^u(\hat{x}) - f^l(\hat{x}) = \sum_{j=1}^{m} c_j^u \hat{x} - \sum_{j=1}^{m} c_j^l \hat{x} = \sum_{j=1}^{m} (c_j^u - c_j^l)\hat{x}$$

Hence to minimize w_f we solve the problem given by (2.1).

2.6 A note on the solution for Objective 4

If the solution obtained for Objective 2 yields a unique solution, then this same solution will satisfy Objective 4. However, if Objective 2 has alternative solutions, then one may get a different solution for Objective 4.

2.7 Selection of the limits used in Objective 5

Let $C = (C^l, C^u)$ be the objective function of some decision problem. If the decision maker is interested in Objective 5 and the limits, L_d and U_d are selected arbitrarily, then there is a danger that the limits will not be satisfied. In the following we derive conditions on L_d and U_d.

Let D be the maximum value of the lower limit the objective function can attain. This could be obtained by maximizing the lower limit. Let E be the minimum value, the upper limit of the objective function can attain. This value can be obtained by solving the problem for Objective 2. Then

$$L_d \leq D \tag{2.2}$$

$$U_d \geq E \tag{2.3}$$

Condition (2.2) and (2.3) are necessary conditions to have a feasible solution. However, they don't guarantee that a solution can be obtained.

Example 2.7.1

..Consider sequencing 3 jobs where the processing times of these jobs are given as $P_1 = [5, 8]$, $P_2 = [6, 10]$ and $P_3 = [4, 9]$. Table 2.1 shows the total flow time for all possible sequences.

Table 2.1: Calculating total flow times for all sequences

Sequences of jobs	Total Flow times	Comments
1-2-3	[31,53]	
1-3-2	[29,52]	Least Upper Limit
2-1-3	[32,55]	Largest Lower Limit
2-3-1	[31,56]	
3-1-2	[28,53]	
3-2-1	[29,57]	

From Table 2.1 we get $(D, E) = (32, 55)$. If $(L_d, U_d) = (30, 55)$ then the sequences 1-2-3 and 2-1-3 satisfy the limits. However, if $(L_d, U_d) = (35, 60)$ then no sequence produces in the limits within this interval..

Chapter 3

Network Problems

3.1 Introduction

In this chapter we consider some network problems and solve them for the interval activity duration times. In Section 3.2 we compute the critical path method where the activity duration times are given as interval values. We also solve the problem of crashing networks and give solutions for two models in Section 3.3. The minimum spanning tree problem, transportation problem and the assignment problems will be solved for different objectives, in Sections 3.4, 3.5 and 3.6 respectively.

3.2 The Interval Critical Path Method

Critical Path Method (CPM) was developed by Kelly and Walker [26] for E.I. du Pond de Nemours Company during late 1950s. To the traditional CPM, we now include the interval approach for the duration estimates. This means that for each activity, instead of having a point estimate we now have lower and upper limits for the activity duration. This will give the project completion time as an interval.

We are considering activity-on-arrow network throughout this chapter. An *event* is a specific point in time which marks the completion time of one or more activities, and is well recognizable in the project [45]. The arrows represent the activities, and nodes represent the events in the network.

We define $[c^l, c^u]$ as the maximum of two interval numbers $[a^l, a^u]$, $[b^l, b^u]$ and is calculated as follows.

$$[c^l, c^u] = max\{[a^l, b^l], [a^u, b^u]\} \tag{3.1}$$

where,

$$c^l = max\{a^l, b^l\} \tag{3.2}$$

$$c^u = max\{a^u, b^u\} \tag{3.3}$$

Similarly the minimum of any two intervals is given by

$$[d^l, d^u] = min\{[a^l, b^l], [a^u, b^u]\} = [min\{a^l, b^l\}, min\{a^u, b^u\}] \tag{3.4}$$

Let $ES_j = [ES_j^l, ES_j^u]$ be the *earliest start time* of all the activities emanating from event j. At the *start* event, we take $j = 1$, then conventionally, for the critical path calculations $[ES_1^l, ES_1^u] = [0,0]$. Let the duration of activity (i,j) be given by $t_{ij} = [t_{ij}^l, t_{ij}^u]$. Let *pred(j)* be the set of events immediately preceding event j. Then, $[ES_j^l, ES_j^u] = max_{i \in pred(j)}\{[t_{ij}^l, t_{ij}^u] + [ES_i^l, ES_i^u]\}$ where,

$$ES_j^l = max_{i \in pred(j)}\{t_{ij}^l + ES_i^l\} \tag{3.5}$$

$$ES_j^u = max_{i \in pred(j)}\{t_{ij}^u + ES_i^u\} \tag{3.6}$$

Let $EF_{ij} = [EF_{ij}^l, EF_{ij}^u]$ be the *earliest finish time* of activity (i,j), where

$$EF_{ij}^l = ES_i^l + t_{ij}^l \tag{3.7}$$

$$EF_{ij}^u = ES_i^u + t_{ij}^u \tag{3.8}$$

Let $LF_i = [LF_i^l, LF_i^u]$ be the *latest finish time* of all the activities coming into event i. Thus, if $j = n$ is the *end* event, then $[LF_n^l, LF_n^u] = [ES_n^l, ES_n^u]$. Let $succ(i)$ be the set of events immediately succeeding event i. Then, by analogy to (3.7) and (3.8) $[LF_i^l, LF_i^u] = min_{j \in succ(i)}\{[LF_j^l - t_{ij}^l, LF_j^u - t_{ij}^u]\}$, where,

$$LF_i^l = min_{j \in succ(i)}\{LF_j^l - t_{ij}^l\} \tag{3.9}$$

$$LF_i^u = min_{j \in succ(i)}\{LF_j^u - t_{ij}^u\} \tag{3.10}$$

The above definition could lead to an anomali as shown in the next example. Consider the network shown in Figure 3.1. The activity times are given on the arrows and the earliest start times for all the activities emanating from any node and latest finish times for all the activities coming into any node are given close to the nodes. The latest finish time of activity $(1,2)$, is calculated as [7-2,14-10] = [5,4]. These values does not satisfy the property of an interval.

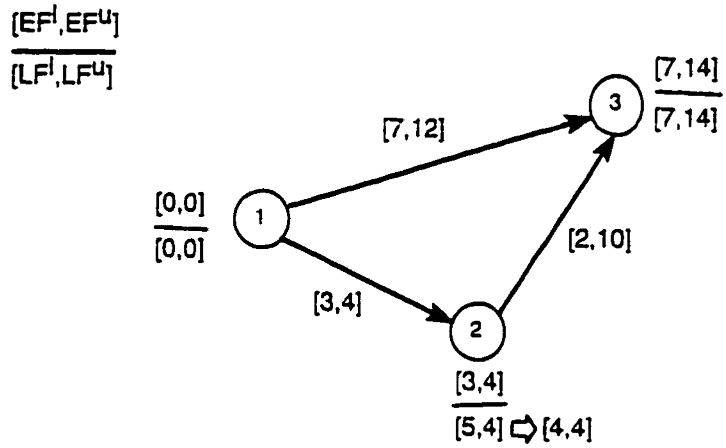

Figure 3.1: Calculating Latest Finish Times

In such a situation we make the lower limit of the latest finish time equal to that of its upper limit. Hence the latest finish times of the activities coming into an event i will be calculated as follows.

$$[LF_i^l, LF_i^u] = [min\{min_{j\in succ(i)}\{LF_j^l - t_{ij}^l\}, LF_i^u\}, LF_i^u] \tag{3.11}$$

For the network considered in Figure 3.1, the latest finish time of all the activities coming into event 2 will now be $[min\{5,4\},4] = [4,4]$.

Let $LS_{ij} = [LS_{ij}^l, LS_{ij}^u]$ be the *latest start time* of an activity (i,j), where

$$LS_{ij}^u = LF_j^u - t_{ij}^u \tag{3.12}$$

$$LS_{ij}^l = min\{LF_j^l - t_{ij}^l, LS_{ij}^u\} \tag{3.13}$$

The *critical activities* can now be identified using the earliest finish times and the latest finish times at the events. An activity (i, j) is *critical* if it satisfies the following conditions.

$$[LF_j^l, LF_j^u] = [LF_i^l, LF_i^u] + [t_{ij}^l, t_{ij}^u] \tag{3.14}$$

$$[ES_j^l, ES_j^u] = [ES_i^l, ES_i^u] + [t_{ij}^l, t_{ij}^u] \tag{3.15}$$

An activity (i, j) is *non-critical* if the following condition is satisfied.

$$EF_{ij}^u < LF_j^l \tag{3.16}$$

The activities which are neither critical nor non-critical are *potentially critical activities.*

Algorithm 3.1 identifies the critical, non-critical and potentially critical activities of a project with interval duration times. The first step is to calculate the earliest start times of all the activities emanating from each event of the project using Equations (3.5) and (3.6). The earliest finish times of each activity is calculated next using Equations (3.7) and (3.8). Fixing the earliest start time of the terminating event as the latest finish time of all the activities coming into the terminating event we calculate the latest finish times of the activities coming into the rest of the events in the network using Equations (3.9) and (3.10) in reverse order. If the lower limit of the latest finish times is found to be greater than the upper limit ($LF_i^l > LF_i^u$) then it is given the value of the upper limit ($LF_i^l = LF_i^u$). Doing this is equivalent to

```
algorithm Critical Activities;
begin
initialize [ES_i^l, ES_i^u] = [0,0],  1 ≤ i ≤ N,
 for j ∈ 2,3,...,N do;
 begin
    ES_j^l = max{t_ij^l + ES_i^l} for all i ∈ pred j ;
    ES_j^u = max{t_ij^u + ES_i^u} for all i ∈ pred j ;
 end for ;
 for k ∈ for all (i,j) activities defined do;
begin
    EF_ij^l = ES_i^l + t_ij^l;
    EF_ij^u = ES_i^u + t_ij^u;
 end for ;
initialize LF_i^l and LF_i^u as large numbers, 1 ≤ i ≤ N,
set LF_N^l = EF_N^l and LF_N^u = EF_N^u;
 for i ∈ N - 1,...,2,1 do;
 begin
    LF_i^l = min{LF_j^l - t_ij^l} for all j ∈ succ i;
    LF_i^u = min{LF_j^u - t_ij^u} for all j ∈ succ i;
    if LF_i^l > LF_i^u then
    LF_i^l = LF_i^u;
 end for ;
 set CA = { }, NCA = { } and PCA = { };
 for k ∈ for all (i,j) activities defined do;
begin
    if [LF_j^l, LF_j^u] = [LF_i^l, LF_i^u] + [t_ij^l, t_ij^u] and
       [ES_j^l, ES_j^u] = [ES_i^l, ES_i^u] + [t_ij^l, t_ij^u] then
     CA = CA U {k};
    else if EF_ij^u < LF_j^l then
     NCA = NCA U {k};
    else PCA = PCA U {k};
    end if
 end for ;
 return CA, NCA and PCA;
end  Critical Activities;
```

Algorithm 3.1: Pseudo code for getting the critical activities for the interval case using Equation (3.11). Three empty sets CA, NCA and PCA are initiated for critical, noncritical and potentially critical activities. Each activity is examined using (3.14), (3.15) and (3.16) whether it is critical, non critical or potentially critical and added into the appropriate set.

In the following we compute the critical path length and identify the critical, non-critical and potentially critical activities for the network shown in Figure 3.2.

Example 3.2.1

Figure 3.2 shows a network using an activity-on-arrow representation and consists of eleven activities. The activity durations are given in the form of intervals as shown on each arrow.

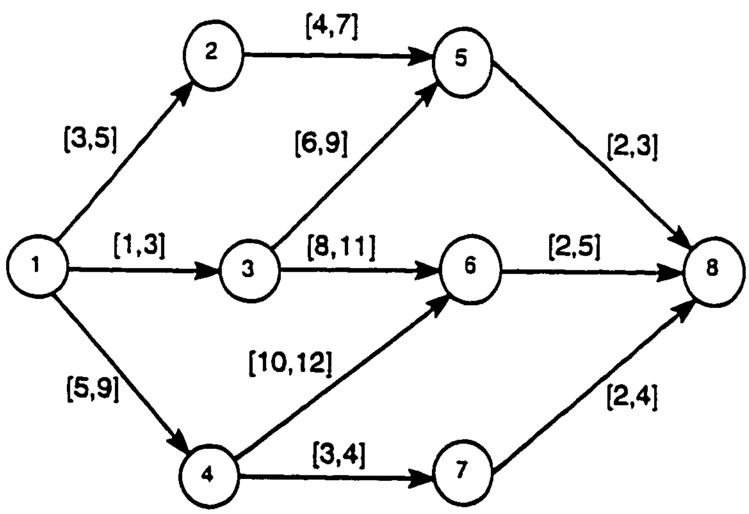

Figure 3.2: Network for Example 3.2.1.

Table 3.1 gives the earliest and latest completion/finish times required for the examination of each activity of Example 3.2.1. The project completion time lies in the interval [17,26]. The critical activities obtained are shown in Figure 3.3 as double lines. In Tables 3.1 and 3.2, C, NC and PC stand for critical, non-critical and potentially critical.

Table 3.1: Examining activities of Example 3.2.1.

Activity (i, j)	duration	ES_i^l	ES_i^u	LF_i^l	LF_i^u	ES_j^l	ES_j^u	LF_j^l	LF_j^u	EF_{ij}^l	EF_{ij}^u	Type
(1, 2)	(3,5)	0	0	0	0	3	5	11	16	3	5	NC
(1, 3)	(1,3)	0	0	0	0	1	3	7	10	1	3	NC
(1, 4)	(5,9)	0	0	0	0	5	9	5	9	5	9	C
(2, 5)	(4,7)	3	5	11	16	7	12	15	23	7	12	NC
(3, 5)	(6,9)	1	3	7	10	7	12	15	23	7	12	NC
(3, 6)	(8,11)	1	3	7	10	15	21	15	21	9	14	NC
(4, 6)	(10,12)	5	9	5	9	15	21	15	21	15	21	C
(4, 7)	(3,4)	5	9	5	9	8	13	15	22	8	13	NC
(5, 8)	(2,3)	7	12	13	18	17	26	17	26	9	15	NC
(6, 8)	(2,5)	15	21	15	21	17	26	17	26	17	26	C
(7, 8)	(2,4)	8	13	15	22	17	26	17	26	10	17	NC

The critical activities for this example are found to be (1,4), (4,6) and (6,8). All the other activities are non-critical. There are no potentially critical activities here, which need not be always true. Example 3.2.1 can be modified to demonstrate this possibility of having potentially critical activities. The duration of the activity (4,7) is changed from (3,4) to (9,14). This will lead to some potentially critical activities as well.

Example 3.2.2

Table 3.2 gives the required earliest finish and latest finish times of the activities.

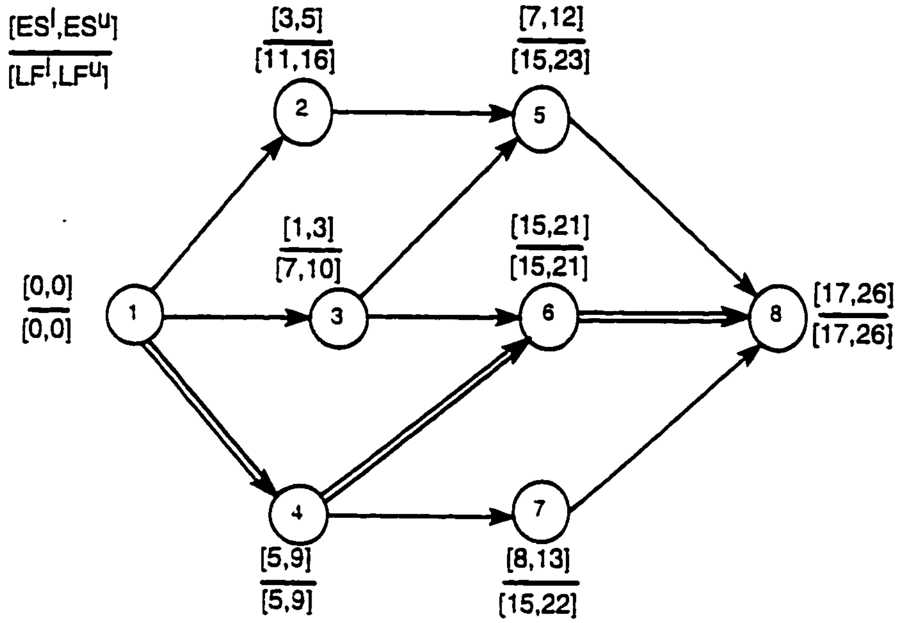

Figure 3.3: Network showing critical activities for Example 3.2.1.

Table 3.2: Examining activities of Example 3.2.2.

Activity (i,j)	duration	ES_i^l	ES_i^u	LF_i^l	LF_i^u	ES_j^l	ES_j^u	LF_j^l	LF_j^u	EF_{ij}^l	EF_{ij}^u	Type
(1, 2)	(3,5)	0	0	0	0	3	5	11	17	3	5	NC
(1, 3)	(1,3)	0	0	0	0	1	3	7	11	1	3	NC
(1, 4)	(5,9)	0	0	0	0	5	9	5	9	5	9	C
(2, 5)	(4,7)	3	5	11	17	7	12	15	24	7	12	NC
(3, 5)	(6,9)	1	3	7	11	7	12	15	24	7	12	NC
(3, 6)	(8,11)	1	3	7	11	15	21	15	22	9	14	NC
(4, 6)	(10,12)	5	9	5	9	15	21	15	22	15	21	PC
(4, 7)	(9,14)	5	9	5	9	14	23	15	23	14	23	PC
(5, 8)	(2,3)	7	12	15	24	17	27	17	27	9	15	NC
(6, 8)	(2,5)	15	21	15	22	17	27	17	27	17	26	PC
(7, 8)	(2,4)	14	23	15	23	17	27	17	27	16	27	PC

The project completion time lies in the interval [17,27]. The critical activities obtained are shown in the Figure 3.4 as double lines and the potentially critical events are shown as dashed lines. The critical activity is (1,4), the non-critical activities

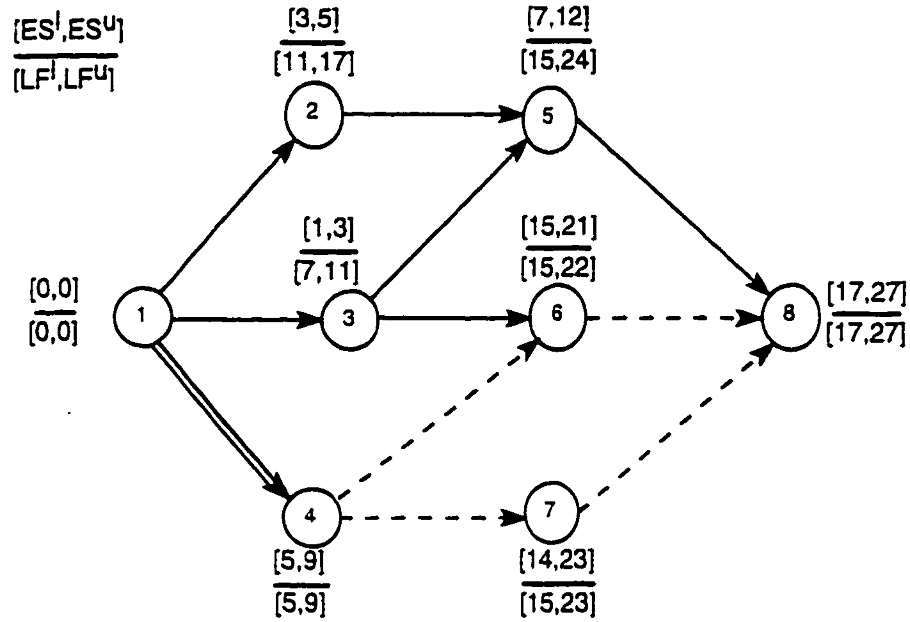

Figure 3.4: Network showing critical activities for Example 3.2.2.

are (1,2), (1,3), (2,5), (3,5), (3,6) and (5,8), and the potentially critical are (4,6), (4,7), (6,8), and (7,8).

Example 3.2.3

This example demonstrates the possibility that a network need not always have a critical activity. In fact all the activities can be potentially critical. Figure 3.5 shows the network. The earliest finish and latest finish times are shown in Figure

3.6. It is clear that all the activities are potentially critical since they are neither critical nor non-critical.

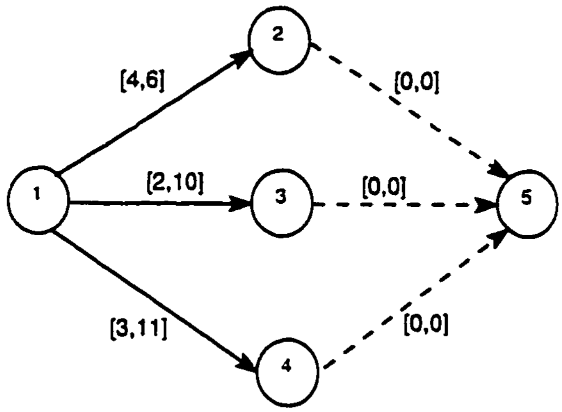

Figure 3.5: Network showing activity times for Example 3.2.3.

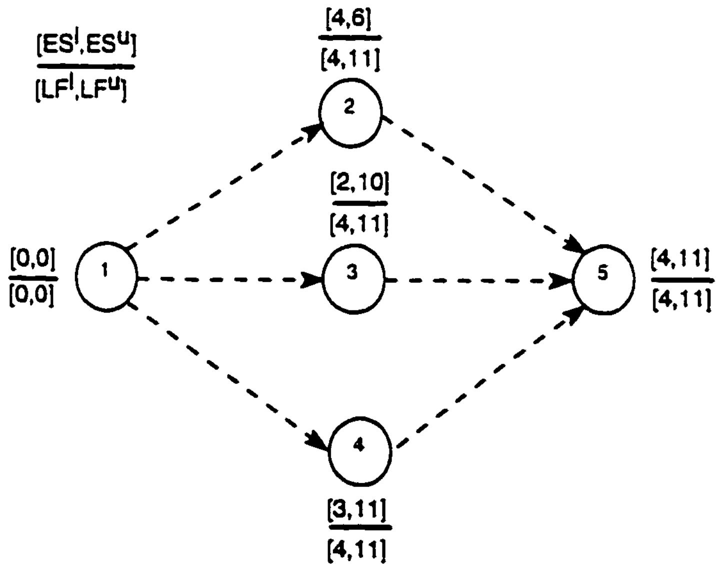

Figure 3.6: Network showing potentially critical activities for Example 3.2.3.

3.2.1 The Decision Makers's Objectives

The project duration is given by the latest finish time of node n, which is computed using (3.9) and ((3.10). The finish time of the project obtained using these equations is the minimum for both the lower and upper limits. So Objective 1 and 2 are satisfied simultaneously. This is also the solution for Objective 4 of minimizing the lower limit given the upper is at its minimum. For this problem Objective 5 becomes impractical. If one chooses an interval, with a higher upper limit, this will not lead him to a smaller lower limit than the one obtained by (3.9). Objective 6 is to obtain a robust schedule and it is solved by Bintong et. al [2].

Objective 3: Minimizing Width of the Project Duration

The width of the project duration can be minimized by allowing its lower limit to increase while maintaining the upper limit of earliest start time of activities emanating from each node. This can be achieved by locating an activity whose delay will lead to the increase in the lower limit only, and maintaining the upper limit constant. The amount of delay should be chosen such that it does not cause an increase in the upper limit of the earliest start time of the activities emanating from any node.

Consider a node j, and let *pred*(j) be the set of nodes preceding j. I.e. (i, j), $i \in pred(j)$ is the set of arrows incident at j. Let $ES_j = [ES_j^l, ES_j^u]$ be the earliest start time of the activities emanating from node j. An activity (i, j), $i \in pred(j)$

can be delayed an amount Δ_{ij} without affecting ES_j^u where

$$\Delta_{ij} = ES_j^u - (ES_i^u + t_{ij}^u) \tag{3.17}$$

Let

$$ES_k^l + t_{kj}^l + \Delta_{kj} = max_{i \in pred(j)}\{ES_i^l + t_{ij}^l + \Delta_{ij}\} \tag{3.18}$$

If activity (k, j) is delayed Δ_{kj}, ES_j^l is maximized while maintaining ES_j^u unchanged.

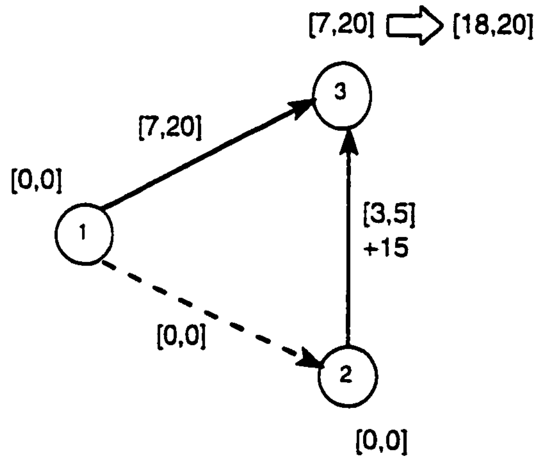

Figure 3.7: Network to demonstrate the method of minimizing the width.

Figure 3.7 displays a network of 3 nodes where the above technique has been demonstrated. Note that $\Delta_{13} = 0$ and $\Delta_{23} = 15$. By delaying activity (2, 3) by 15 time units we get a larger lower limit of the earliest finish time for arrows incident to node 3. The width is thus reduced from $20 - 7 = 13$ to $20 - 18 = 2$. For larger networks this method can easily be applied to obtain the minimum width of the project completion time.

Algorithm 3.2 implements the above method. This algorithm will give the modified earliest start times and the activities which are selected with the amount of increment. In the algorithm $(ij)^*$ is used as the index to identify the activity selected and *inc* is the amount of increment.

```
algorithm Minimizing Width in CPM;
begin
initialize ES_i^l = 0 and ES_i^u = 0, inc = 0, 1 <= i <= N,
 for j ∈ 2,3,...,n do;
  begin
  ES_j^l = max_{i∈pred_j} {t_ij^l + ES_i^l};
  ES_j^u = max_{i∈pred_j} {t_ij^u + ES_i^u};
    for i ∈ pred_j do;
     begin
       Δ_ij = ES_j^u - (ES_i^u + t_ij^u);
       L_ij = ES_i^l + t_ij^l + Δ_ij;
    end for ;
    (ij)* = arg max_{i∈pred_j} {L_ij};
    inc = Δ_(ij)*;
    ES_j^l = ES_i^l + t_(ij)*^l + inc;
    Return ES_j^l, ES_j^u, (ij)*, inc;
 end for ;
end algorithm;
```

Algorithm 3.2: Pseudo code for getting the minimum width for CPM

Example 3.3.1

In this example we consider a network with 13 activities and 8 events. The network is shown in Figure 3.8. The earliest start times and latest finish times of activities are given next to each node.

Only one critical activity (1,4), is found in this project. Activities (1,2), (2,3)

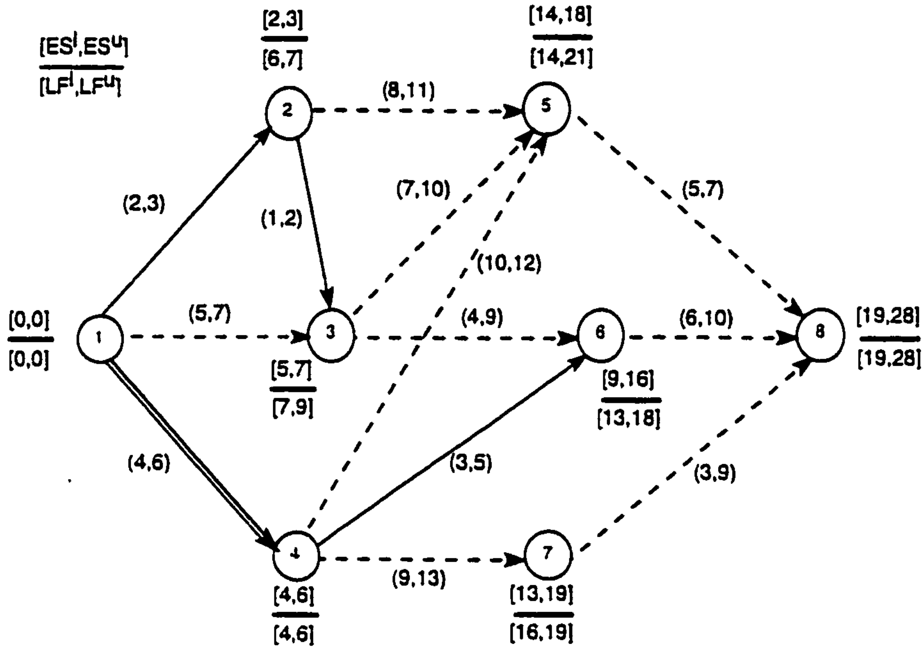

Figure 3.8: Network for Example 3.3.2. (Before applying Algorithm 3.2)

and (4.6) are found to be non-critical. All the remaining activities are potentially critical activities. The project completion time is [19.28] and hence the width of this interval is 9. Now Algorithm 3.2 is applied to this network to minimize the width of this project completion time interval. Figure 3.9 gives the network after applying Algorithm 3.2.

Figure 3.9 shows the amount of delay possible. If an activity is selected to be delayed then the increment is given after the + sign. The critical activities are shown as double lines, the potentially critical activities as dashed lines and the non-critical as thin lines in both the Figures 3.8 and 3.9.

The project completion time calculated after applying the algorithm is [22.28]

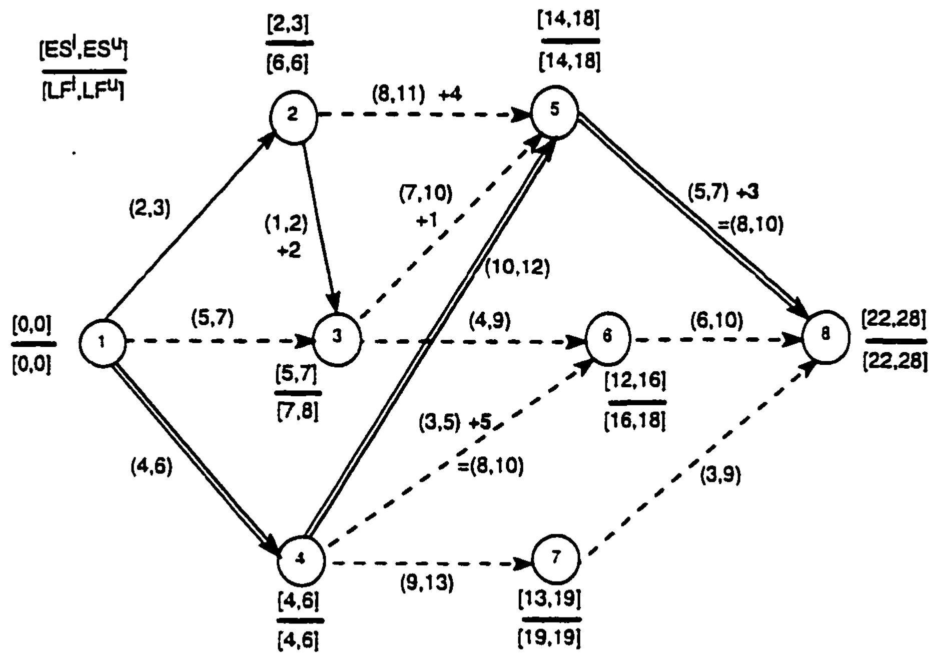

Figure 3.9: Network for Example 3.3.2. (After applying Algorithm 3.2)

and now the width is 6. The critical activities are (1,4), (4,5) and (5,8). The non-critical activities are (1,2) and (2,3), and the remaining activities are potentially critical. This example shows that by delaying some activities the criticality of these activities is increased. If a non-critical activity is delayed, it may become a potentially critical or critical activity, and if a potentially critical activity is delayed then it will turn into a critical activity. A critical activity can never be delayed because, it will effect the project completion time, however its completion time marks the limit unto which one of the activities coming into the same node can be delayed.

In this example activity (4,6), a non-critical activity was delayed by 5 time units and it turned into a potentially critical activity. Activity (5,8) was a potentially critical activity and it turned into a critical activity when delayed for 3 time units.

3.3 Crashing Networks

In crashing networks it is assumed that by assigning additional resources to an activity, its duration can be reduced to an certain extent. The additional cost incurred in reducing the activity time is called the crashing cost.

The total cost of a project is the sum of the direct costs (inversely proportional to the activity times) and the indirect costs (proportional to the project duration). By crashing the critical activities, the project duration can be reduced, but of course the total cost of the project will increase. In crashing projects we consider the trade-off between the total cost of the project and its completion time. The project crashing problem is to determine the amount by which the various activities are to be crashed when a project is required to be completed within a given time, or when a limited budget is assigned.

We now discuss the critical path analysis and it is assumed that the cost-versus-time relationship is available for every activity in the project and apply the interval approach in the crashing of the networks. First, consider network crashing given point activity times.

Let t_{ij} be duration time of activity (i, j) and t_i be the unknown event times $(1 \leq i \leq n)$ hence $t_j - t_i \geq t_{ij}$.

Let n_{ij} be normal duration time of activity (i, j) when no additional resources are assigned, and c_{ij} be crash duration time of activity (i, j) with maximum amount of resources, hence $c_{ij} \leq t_{ij} \leq n_{ij}$.

Let C_{ij} represent the cost of shortening the duration of activity (i, j) by one time unit. The cost of crashing activity (i, j) is given by $C_{ij}(n_{ij} - t_{ij})$.

Next consider the case where an activity duration is given as intervals. If no additional resources are assigned then the normal duration will fall in an interval $[n_{ij}^l, n_{ij}^u]$. On the other hand if activity (i, j) is crashed by allocating maximum amount of resources then its duration will fall in the interval $[c_{ij}^l, c_{ij}^u]$. Let the activity duration be $t_{ij} = [t_{ij}^l, t_{ij}^u]$. Let Z_{ij}^n and Z_{ij}^c be the cost of completing the activity at its normal duration time and its crashed time respectively.

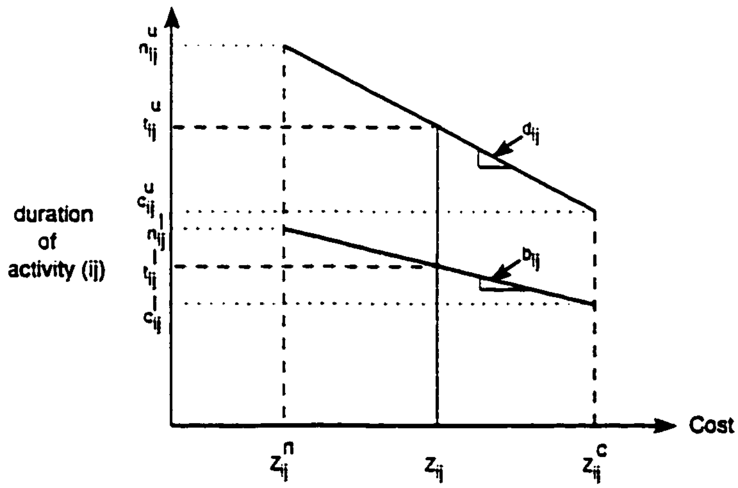

Figure 3.10: The Durations of activity (i, j) versus Cost graph

The duration of activity (i, j) versus the cost graph is shown in Figure 3.10, where the duration of activity (i, j) is given as an interval. At a given crashing cost Z_{ij}, the activity duration t_{ij} falls in the interval $[t^l_{ij}, t^u_{ij}]$. We define d_{ij} as the change in activity duration per unit change in cost when the activities are at upper limit and b_{ij} is the change in activity duration per unit change in cost when the activities are at lower limit. I.e.,

$$d_{ij} = \frac{n^u_{ij} - c^u_{ij}}{Z^c_{ij} - Z^n_{ij}} \tag{3.19}$$

$$b_{ij} = \frac{n^l_{ij} - c^l_{ij}}{Z^c_{ij} - Z^n_{ij}} \tag{3.20}$$

We assume that the uncertainty in the activity duration decreases as cost of crashing increases. i.e.

$$n^u_{ij} - n^l_{ij} \geq c^u_{ij} - c^l_{ij} \tag{3.21}$$

Dividing both sides of (3.21) by $(Z^c_{ij} - Z^n_{ij})$, we obtain after rearranging terms

$$d_{ij} = \frac{n^u_{ij} - c^u_{ij}}{Z^c_{ij} - Z^n_{ij}} \geq \frac{n^l_{ij} - c^l_{ij}}{Z^c_{ij} - Z^n_{ij}} = b_{ij} \tag{3.22}$$

The values of t^l_{ij} and t^u_{ij} are given by

$$t^l_{ij} = (Z^c_{ij} - Z_{ij})b_{ij} + c^l_{ij} \tag{3.23}$$

$$t^u_{ij} = (Z^c_{ij} - Z_{ij})d_{ij} + c^u_{ij} \tag{3.24}$$

We formulate the network crashing problem as a linear program and solve for two models. Irrespective of the model being solved the following five constraints will always be included in the linear program.

$$[c^l_{ij}, c^u_{ij}] \leq [t^l_{ij}, t^u_{ij}] \leq [n^l_{ij}, n^u_{ij}] \tag{3.25}$$

$$Z_{ij} = Z_{ij}^{c} - \frac{1}{d_{ij}}(t_{ij}^{u} - c_{ij}^{u}) \tag{3.26}$$

$$t_j^l - t_i^l \geq t_{ij}^l \tag{3.27}$$

$$t_j^u - t_i^u \geq t_{ij}^u \tag{3.28}$$

$$\frac{1}{d_{ij}}(t_{ij}^{u} - c_{ij}^{u}) = \frac{1}{b_{ij}}(t_{ij}^{l} - c_{ij}^{l}) \tag{3.29}$$

Model I

Given that the project must be completed by time $T = [T^l, T^u]$, determine how the project activities are to be expedited such that the total crashing cost is minimized.

The objective function of the linear program to be solved will be *minimize* $\sum_{(ij)\in A}^{n} Z_{ij}$, where A is the set of all the activities in the network. An additional constraint $T^l \leq t_n^l \leq t_n^u \leq T^u$ should be added to the five constraints from (3.25) to (3.29) to solve for model I.

Example 3.3.1

The network for this example is shown in Figure 3.11. The figure shows the normal and crashed duration times. The data for normal and crashed cost for each activity is given in Table 3.3. The objective is to crash the activities such that the project is completed within the time interval [13,20]. The solution obtained for this problem is shown in Table 3.4. The total direct cost involved after crashing the activities is 1290.

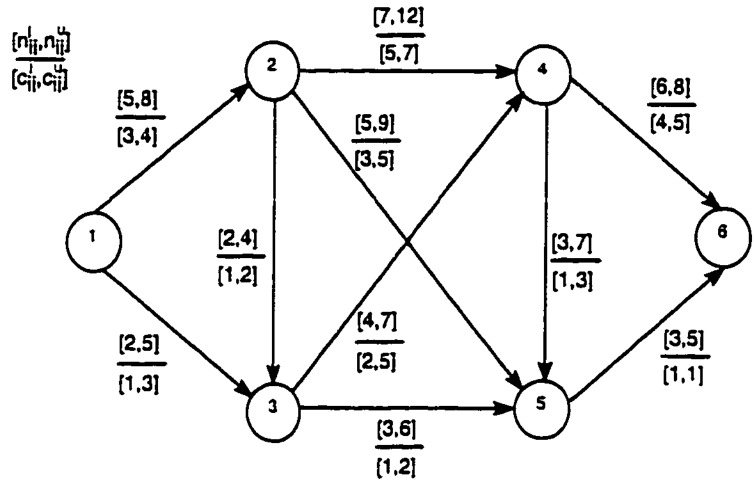

Figure 3.11: Network for Example 3.3.1.

Table 3.3: Data for Example 3.3.1

Activity	Normal			Crash				
	n^l_{ij}	n^u_{ij}	Z^n_{ij}	c^l_{ij}	c^u_{ij}	Z^c_{ij}	$1/b_{ij}$	$1/d_{ij}$
1-2	5	8	100	3	4	200	50	25
1-3	2	5	50	1	3	80	30	15
2-3	2	4	150	1	2	180	30	15
2-4	7	12	200	5	7	250	25	10
2-5	5	9	60	3	5	80	10	5
3-4	4	7	80	2	5	100	10	10
3-5	3	6	60	1	2	80	10	5
4-5	3	7	140	1	3	160	10	5
4-6	6	8	150	4	5	180	15	10
5-6	3	5	120	1	1	160	20	10

The linear program for the example will now be as follows.

$$Min \quad Z_{12} + Z_{13} + Z_{23} + Z_{24} + Z_{25} + Z_{34} + Z_{35} + Z_{45} + Z_{46} + Z_{56}$$

subject to

$$Z_{12} + 25t^u_{12} = 300$$
$$Z_{13} + 15t^u_{13} = 125$$
$$Z_{23} + 15t^u_{23} = 210$$
$$Z_{24} + 10t^u_{24} = 320$$
$$Z_{25} + 5t^u_{25} = 105$$
$$Z_{34} + 10t^u_{34} = 150$$
$$Z_{35} + 5t^u_{35} = 90$$
$$Z_{45} + 5t^u_{45} = 175$$
$$Z_{46} + 10t^u_{46} = 230$$
$$Z_{56} + 10t^u_{56} = 170$$
$$t^u_{12} - 2t^l_{12} = -2$$
$$t^u_{13} - 2t^l_{13} = 1$$
$$t^u_{23} - 2t^l_{23} = 0$$
$$2t^u_{24} - 5t^l_{24} = -11$$
$$t^u_{25} - 2t^l_{25} = -1$$
$$t^u_{34} - t^l_{34} = 3$$
$$t^u_{35} - t^l_{35} = 0$$
$$t^u_{45} - 2t^l_{45} = 1$$
$$2t^u_{46} - 3t^l_{46} = -2$$
$$t^u_{56} - 2t^l_{56} = -1$$
$$3 \leq t^l_{12} \leq 5$$
$$4 \leq t^u_{12} \leq 8$$
$$1 \leq t^l_{13} \leq 2$$
$$3 \leq t^u_{13} \leq 5$$
$$1 \leq t^l_{23} \leq 2$$
$$2 \leq t^u_{23} \leq 4$$

$$5 \le t^l_{24} \le 7$$
$$7 \le t^u_{24} \le 12$$
$$3 \le t^l_{25} \le 5$$
$$5 \le t^u_{25} \le 9$$
$$2 \le t^l_{34} \le 4$$
$$5 \le t^u_{34} \le 7$$
$$1 \le t^l_{35} \le 3$$
$$2 \le t^u_{35} \le 6$$
$$1 \le t^l_{45} \le 3$$
$$3 \le t^u_{45} \le 7$$
$$4 \le t^l_{46} \le 6$$
$$5 \le t^u_{46} \le 8$$
$$1 \le t^l_{56} \le 3$$
$$1 \le t^u_{56} \le 5$$
$$t^l_2 - t^l_1 \ge t^l_{12}$$
$$t^u_2 - t^u_1 \ge t^u_{12}$$
$$t^l_3 - t^l_1 \ge t^l_{13}$$
$$t^u_3 - t^u_1 \ge t^u_{13}$$
$$t^l_3 - t^l_2 \ge t^l_{23}$$
$$t^u_3 - t^u_2 \ge t^u_{23}$$
$$t^l_4 - t^l_2 \ge t^l_{24}$$
$$t^u_4 - t^u_2 \ge t^u_{24}$$
$$t^l_4 - t^l_3 \ge t^l_{34}$$
$$t^u_4 - t^u_3 \ge t^u_{34}$$
$$t^l_5 - t^l_2 \ge t^l_{25}$$
$$t^u_5 - t^u_2 \ge t^u_{25}$$
$$t^l_5 - t^l_3 \ge t^l_{35}$$
$$t^u_5 - t^u_3 \ge t^u_{35}$$
$$t^l_5 - t^l_4 \ge t^l_{45}$$

$$t_5^u - t_4^u \geq t_{45}^u$$
$$t_6^l - t_4^l \geq t_{46}^l$$
$$t_6^u - t_4^u \geq t_{46}^u$$
$$t_6^l - t_5^l \geq t_{56}^l$$
$$t_6^u - t_5^u \geq t_{56}^u$$
$$t_6^l \geq 13$$
$$t_6^u \leq 20$$
$$t_6^l \leq t_6^u$$

Table 3.4: Solution for Example 3.3.1

Activity	Duration	Comment
1-2	[4,6]	Crashed
1-3	[2,5]	Normal
2-3	[1,2]	Normal
2-4	[5.8,9]	Crashed
2-5	[5,9]	Normal
3-4	[2,5]	Crashed
3-5	[3,6]	Normal
4-5	[1,3]	Crashed
4-6	[4,5]	Crashed
5-6	[1.5,2]	Crashed

Model II

If cost due to crashing the project activities and is to be limited to budget B, determine how this budget can be allocated such that the project completion time is minimized.

Since the objective of this model is to minimize project duration which will fall in an interval $[t_n^l, t_n^u]$, this model can be solved for the five objectives of the decision maker. We will consider each objective and show how they can be achieved.

This is accomplished by solving the linear program with different objective functions for different objectives of the decision maker. A constraint $\sum_{(ij)\in A} Z_{ij} \leq B$ should be added to the set of constraints form (3.25) to (3.29).

Objectives 1 and 2

Objectives 1 and 2 are to minimize the lower limit t_n^l and the upper limit t_n^u of the project completion time respectively. Objectives 1 and 2 can be achieved by giving the objective function as t_n^l and t_n^u respectively.

Consider the network given in Figure 3.11. The normal and crashable durations of each activity is given in Table 3.3. The solution for Objective 1 is shown in the Table 3.5. The project duration is found to be [14,26].

The solution obtained for Objective 2 for the same network is shown in Table 3.6. The project completion time is found to be [15.47,23].

Objective 3

Objective 3 is to minimize the width $t_n^u - t_n^l$ of the project completion time interval. If $t_n^u - t_n^l$ is given as the objective function in the linear program, then the solution will have $t_n^u = t_n^l$ since there is no limit on t_n^l.

Table 3.5: Solution for Objective 1

Activity	Duration	Comment
1-2	[5,8]	Normal
1-3	[2,5]	Normal
2-3	[2,4]	Normal
2-4	[5,7]	Crashed
2-5	[5,9]	Normal
3-4	[3,6]	Crashed
3-5	[3,6]	Normal
4-5	[1,3]	Crashed
4-6	[4,5]	Crashed
5-6	[3,5]	Normal

Table 3.6: Solution for Objective 2

Activity	Duration	Comment
1-2	[5,8]	Normal
1-3	[2,5]	Normal
2-3	[2,4]	Normal
2-4	[5.8,9]	Crashed
2-5	[5,9]	Normal
3-4	[2,5]	Crashed
3-5	[3,6]	Normal
4-5	[1,3]	Crashed
4-6	[4.67,6]	Crashed
5-6	[2,3]	Crashed

To achieve the minimum width for the project completion time we propose a heuristic which has two steps. The first step is to minimize the upper limit of the project time for a given budget, Objective 2. This will not minimize the width of the project completion time. In step 2 we apply Algorithm 3.2 on crashed activity times. Algorithm 3.2 will select some of the activities within the network, and delay their start to minimize the width of the project completion time (refer Section 3.2.1).

Example 3.3.2

Consider the network given in Figure 3.11. The data for the normal times and the crashed times of the activities are given in Table 3.3.

The project completion time when this problem is solved for Objective 2 is [15.47,23]. The solution is given in Table 3.6. The width of the project completion time is 7.53. We now apply Algorithm 3.2 to this network. It is found that activities (1,3) and (3,5) can be delayed by 7 and 5 time units respectively. Now the project completion time will become [16,23] and the width is reduced to 7 units. The solution obtained after applying the algorithm is given in Figure 3.12

Objective 4

Objective 4 is to minimize lower limit of the project completion time given that the upper limit is minimized. To achieve this objective we have to add a constraint to

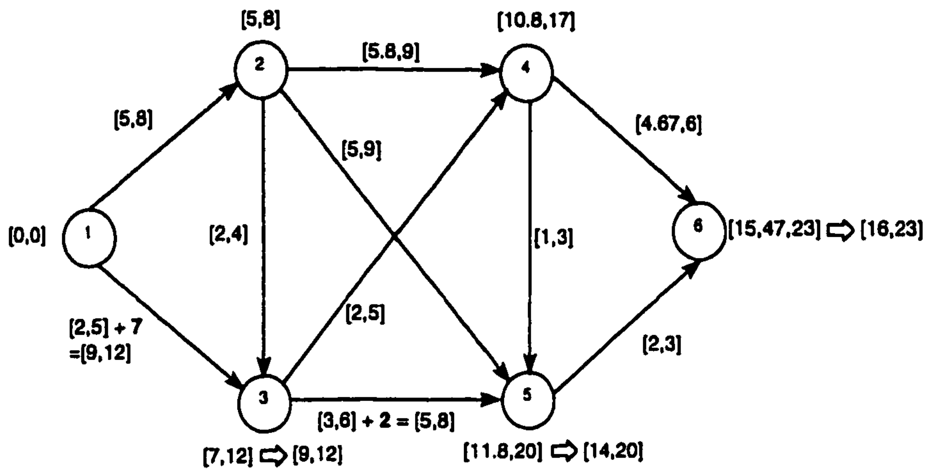

Figure 3.12: Solution for Example 3.3.2.

the linear program for Objective 2 of Model II. This constraint is given by;

$$t_n^u \leq T^*$$

where T^* is the solution of Objective 2 which is to minimize the upper limit of the project completion time. If a problem has unique solution for Objective 2, then the solution for Objective 4 will also be same. If alternative solution exists then we will get a solution with better lower limit.

Example 3.3.3

Consider the network given in Figure 3.11. The constraint which has to be added is $t_6^u \leq 23$ where we use the solution of Objective 2 as the upper limit of the project completion time. For this example with budget of 1220 we have obtained a solution [15.2,23], which is different from Objective 2 since there is an alternative solution. The solution obtained is shown in Table 3.7.

Table 3.7: Solution for Objective 4

Activity	Duration	Comment
1-2	[5,8]	Normal
1-3	[2,5]	Normal
2-3	[2,4]	Normal
2-4	[6.2,10]	Crashed
2-5	[5,9]	Normal
3-4	[3,6]	Crashed
3-5	[3,6]	Normal
4-5	[1,3]	Crashed
4-6	[4,5]	Crashed
5-6	[1.5,2]	Crashed

Objective 5

Objective 5 is to obtain a solution between desired interval $[d^l, d^u]$. In the case of Model II where we are assigning additional cost to reduce the project completion time by crashing some critical activities, the minimum value d^u can take can be obtained by solving for Objective 2. Let the solution obtained for Objective 2 be $[T^l, T^u]$. Two constraints $t_n^l \geq d^l$ and $t_n^u \leq d^u$ should be added to the linear program of Objective 2. If $d^u < T^u$ is given as the constraint then there will be no feasible solution for Objective 5. If $d^l \leq T^l$ and $d^u \geq T^u$, then we will obtain the same solution of Objective 2, for Objective 5. If $d^l \geq T^l$ and $d^u \geq T^u$ then the linear program will simply increase t_n^l to the desired lower limit d^l and t_n^u will be the same as obtained for Objective 2.

3.4 Minimal Spanning Tree Problem

Consider an undirected connected network $G = (N, A)$ where $N = \{1, 2, .., m\}$ is a set of m nodes and $A = \{(ij) \mid i < j,\ i, j \ \in N\}$ is a set of arrows. Associated with each arrow $(ij) \in A$ is a cost c_{ij}.

A *tree* is a connected network that may involve a subset of nodes and has no loops. A *spanning tree* is a tree that includes all nodes.

A *Minimal spanning tree* is a spanning tree such that $\sum_{(ij) \in S \subset A} c_{ij}$ is minimized, where S is to be determined.

The minimum spanning tree problem can be solved via a "*greedy approach*" [18] which requires minimum amount of effort. Using this "*greedy*" rationale the following is the algorithm for the general case.

Step 1: Define two sets of nodes using the original list of nodes.
a. S = a set of connected nodes.
b. $\overline{S}$ = a set of unconnected nodes.
Initially all the nodes are will be placed in set $\overline{S}$.
Step 2: Choose any node in $\overline{S}$ and connect it to its closest node.
(This step will place two nodes initially in the set S.)
Step 3: Connect the node from $\overline{S}$ which is closest to any node in set S; call the node that has been selected from set $\overline{S}$,δ. Transfer δ from $\overline{S}$ to set S.
Step 4: Repeat Step 3 until all nodes are in set S.

Algorithm 3.3: Algorithm based on Greedy Approach for the STP

Now assume that the cost $c_{ij} = [c^l_{ij}, c^u_{ij}]$. In the following we solve this problem for different Objectives.

Objectives 1, 2 and 3

Algorithm 3.3 can be used to achieve Objectives 1, 2 and 3 by using for each arc, the values c_{ij}^l, c_{ij}^u and widths $w_{ij} = (c_{ij}^u - c_{ij}^l)$ respectively.

Theorem 3.4.1:

Minimizing the variation (width) in the objective function of a spanning tree problem is achieved by solving it using the widths w_{ij} as the cost values.

Proof:

Let S be the set of arrows in a spanning tree which are not necessarily optimal, then the cost is given by

$$C_S = [C^l, C^u] = \sum_{ij\epsilon S}[c_{ij}^l, c_{ij}^u] = [\sum_{ij\epsilon S} c_{ij}^l, \sum_{ij\epsilon S} c_{ij}^u]$$

The width of C_S is given by

$$W_S = \sum_{ij\epsilon S}(c_{ij}^u - c_{ij}^l) = \sum_{ij\epsilon S} w_{ij}$$

Hence to minimize W_S one uses Algorithm 3.3 with cost w_{ij} for arc (i,j).

Objective 4: Minimizing c_{ij}^l given c_{ij}^u is minimized

To achieve this objective we minimize the upper limit first which is Objective 2 of the decision maker. In case of a tie we choose the arc with the smaller lower limit value. In this way we are minimizing the lower limit while making sure that the upper limit is at its minimum. If there are no ties while solving for Objective 2 then the solution for Objectives 2 and 4 will be identical.

Example 3.4.1

Consider the network given in Figure 3.13. The cost assigned with each arrow is given next to the arc. The solution obtained after applying the Greedy Algorithm and breaking ties for the upper limits by considering the lower limit of the cost we obtain the spanning tree shown in Figure 3.14.

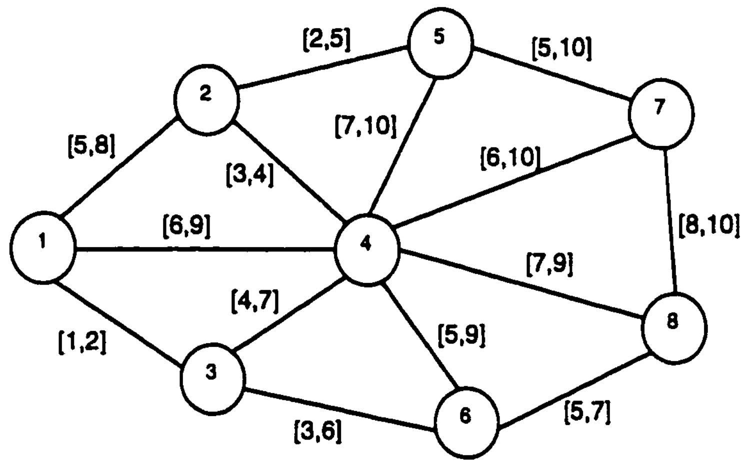

Figure 3.13: Network for Example 3.4.1.

Table 3.8 summarizes the steps involved while applying the Greedy algorithm for this example. The last column lists the arcs if there are ties. The arc selected after breaking ties with respect to the lower limit of the costs are marked with a $\star$. The optimal cost is found to be [23,41].

Table 3.8: Applying Greedy Algorithm for Example 3.4.1

Iteration	S	Node Selected	Tie Between arcs
0	{ ϕ }	1	-
1	{ 1 }	3	-
2	{ 1,3 }	6	-
3	{ 1,3,6 }	4	(3,4)* & (6,8)
4	{ 1,3,4,6 }	2	-
5	{ 1,2,3,4,6 }	5	-
6	{ 1,2,3,4,5,6 }	8	-
7	{ 1,2,3,4,5,6,8 }	7	(4,7), (5,7)* & (7,8)
8	{ 1,2,3,4,5,6,7,!8 }	-	-

* selected arc.

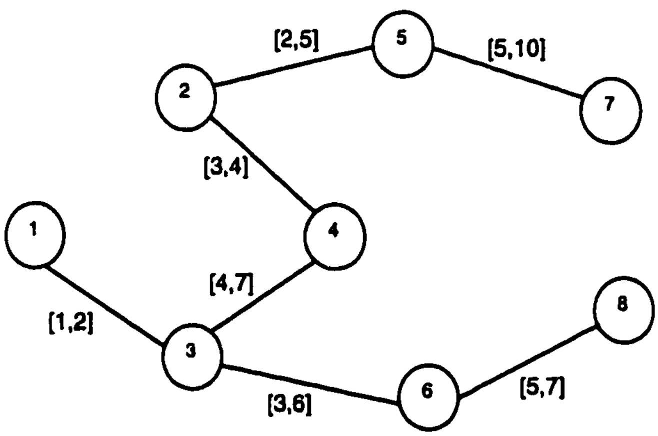

Figure 3.14: Solution for Example 3.4.1.

Objective 5:

Obtaining a solution within desired limits can be tried by starting with any initial network and improving upon the solution by replacing arrows. There is a danger that this method could lead us to a local minimum. Let us consider the network given in Figure 3.15 as some intermediate solution with total cost [32,48]. A solution within the limits [33,48] is desired. It is known that there is an optimal solution within this limits. To obtain this objective we have to replace some arrow which will improve the solution. In the given network we can see that if we try to replace any

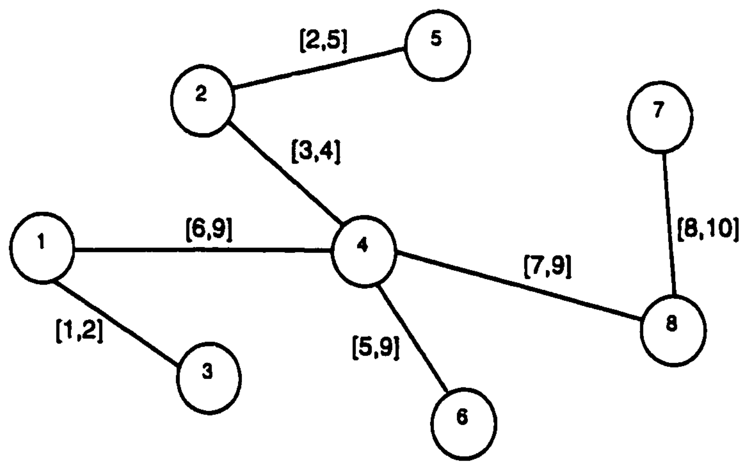

Figure 3.15: An Observation for Objective 5

arrow with another arrow it will either lead to increased upper limit than desired or decrease the lower limit. Hence, we are struck in a local minimum while we know that there is an optimal solution within these limits.

The problem of finding the minimal cost spanning tree can be formulated as an integer program as shown below. The solution time of the resulting integer program could be time consuming for large problems.

IP formulation for Spanning Tree Problem

Let,

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

The integer program is given by

$$Minimize \sum_{(ij)\in A} c_{ij}x_{ij}$$

$$\sum_{(ij)\in A} x_{ij} + \sum_{(ji)\in A} x_{ji} \geq 1 \quad for\ all\ i \in N,$$

$$\sum_{(ij)\in A} x_{ij} = m - 1$$

$$x_{ij} = 0\ or\ 1$$

In this formulation we assume that $c_{ij} > 0$ for all arrows (i,j). The first set of constraints guarantee that each node is connected to at least one other node. The next constraint implements a spanning tree condition that the number of arcs is exactly equal to number of nodes -1. If the solution obtained from the above formulation is found to have cycle, we add constraint (3.30) given below to make sure that the solution obtained is a spanning tree.

Let $A(S)$ denote the set of arrows forming a cycle which contains the set of nodes, S. The following constraint should be added to the integer program.

$$\sum_{(ij)\in A(S)} x_{ij} = |S| - 1 \tag{3.30}$$

Where $|S|$ is the number of nodes in the set S. It is possible that the new solution obtained may still have a cycle. In this case we repeat the procedure of adding a constraint and continue till we get a spanning tree.

To obtain Objective 5 for the minimal spanning tree problem, we have to find a feasible integer solution to the system given below. The conditions on the selection desired interval $[L_d, U_d]$ are given in Section 2.3.

$$\sum_{(ij)\in A} x_{(ij)} + \sum_{(ji)\in A} x_{ji} \geq 1 \quad 1 \leq i \leq N,$$

$$\sum_{(ij)\in A} x_{ij} = m - 1$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c^{l}_{ij} x_{ij} \geq L_d$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c^{u}_{ij} x_{ij} \leq U_d$$

$$x_{ij} = 0 \text{ or } 1$$

We have to add the constraint (3.30) and solve the integer program if the solution contains a cycle.

Example 3.4.2

Consider a network shown in Figure 3.13 for Example 3.4.1. The system developed for this data is given below. In this example we want to get a solution in the interval [33,48].

$$x_{12} + x_{13} + x_{14} \geq 1$$

$$x_{12} + x_{24} + x_{25} \geq 1$$

$$x_{13} + x_{34} + x_{36} \geq 1$$

$$x_{14} + x_{24} + x_{34} + x_{45} + x_{46} + x_{47} + x_{48} \geq 1$$

$$x_{25} + x_{45} + x_{57} \geq 1$$

$$x_{36} + x_{46} + x_{68} \geq 1$$

$$x_{47} + x_{57} + x_{78} \geq 1$$

$$x_{48} + x_{68} + x_{78} \geq 1$$

$$x_{12} + x_{13} + x_{14} + x_{24} + x_{25} + x_{34} + x_{36} +$$
$$x_{45} + x_{46} + x_{47} + x_{48} + x_{57} + x_{68} + x_{78} = 7$$

$$5x_{12} + x_{13} + 6x_{14} + 3x_{24} + 2x_{25} + 4x_{34} + 3x_{36} + 7x_{45}$$
$$+5x_{46} + 6x_{47} + 7x_{48} + 5x_{57} + 5x_{68} + 8x_{78} \geq 33$$

$$8x_{12} + 2x_{13} + 9x_{14} + 4x_{24} + 5x_{25} + 7x_{34} + 6x_{36} + 10x_{45}$$
$$+9x_{46} + 10x_{47} + 9x_{48} + 10x_{57} + 7x_{68} + 10x_{78} \geq 48$$

$$x_{ij} = 0 \; or \; 1$$

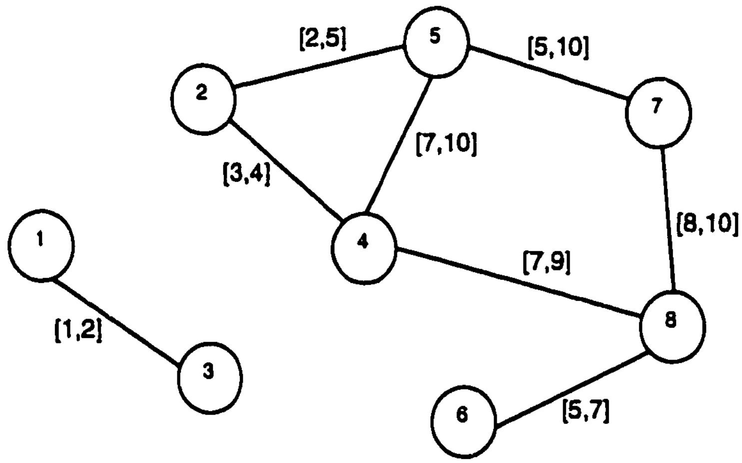

Figure 3.16: First solution for Example 3.4.2.

The solution obtained for this system is shown in Figure 3.16. It can be seen that there is a cycle which is causing network to be disconnected and hence we add the following constraint to avoid this to happen.

$$x_{24} + x_{25} + x_{45} = 2$$

The solution obtained after adding this constraint is shown in Figure 3.17. The objective value is found to be [34,48] as desired.

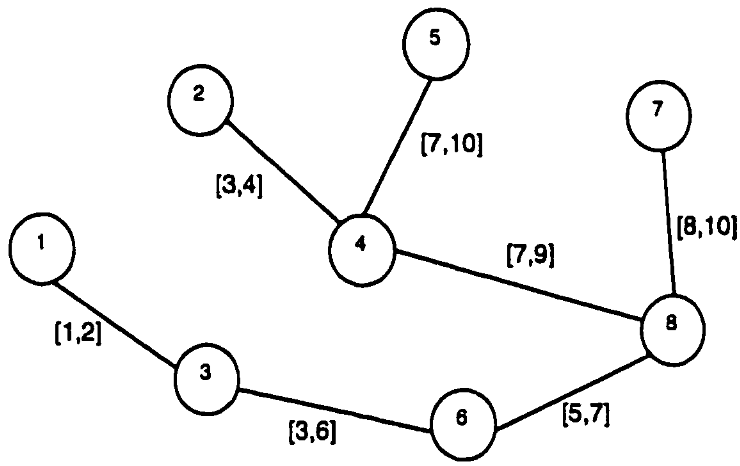

Figure 3.17: Optimal solution for Example 3.4.2.

3.5 Transportation Problem (TP)

Definition :

Consider m suppliers, where supplier i has a supply of a_i units of a particular commodity. In addition, there are n destinations, where destination j requires b_j units. Let C_{ij} be the cost of transporting one unit from supplier i to the destination j. The problem is to determine the feasible shipping pattern from suppliers to destinations that minimizes the total transportation cost.

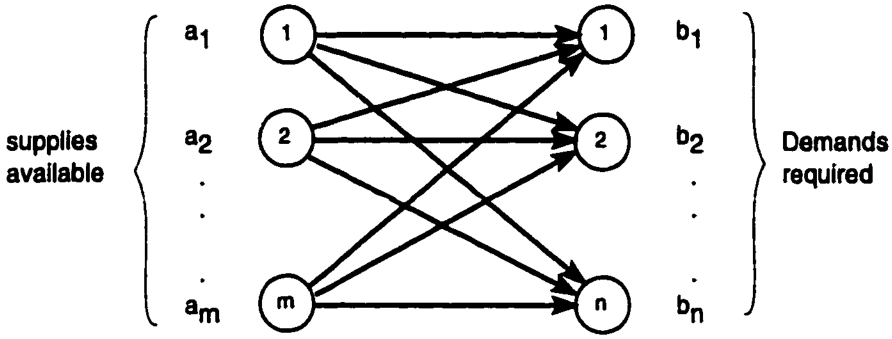

Figure 3.18: Graphical structure of the Transportation Problem

The well-known transportation problem is formulated as follows. Let x_{ij} be the number of units shipped from supplier i to destination j. The linear program for this transportation problem is given below.

LP for general TP

$$Minimize \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} \leq a_i \quad 1 \leq i \leq m \tag{3.31}$$

$$\sum_{i=1}^{m} x_{ij} \geq b_i \quad 1 \leq j \leq n \tag{3.32}$$

$$x_{ij} \geq 0$$

Interval Cost Variables

We now consider the transportation problem where the cost values fall in an interval $[c^l_{ij}, c^u_{ij}]$. We can now solve the TP for different objectives.

Objectives 1, 2 and 3

The first three objectives of the decision maker can be achieved by solving the linear program given below.

$$Minimize \sum_{i=1}^{m}\sum_{j=1}^{n} \mathrm{C}_{ij}x_{ij}$$

subject to

$$\sum_{j=1}^{n} x_{ij} \leq a_i \quad 1 \leq i \leq m$$

$$\sum_{i=1}^{m} x_{ij} \geq b_i \quad 1 \leq j \leq n$$

$$x_{ij} \geq 0$$

To achieve Objective 1, 2 or 3 (refer Section 2.5) we take C_{ij} as c^l_{ij}, c^u_{ij} or $c^u_{ij} - c^l_{ij}$ respectively, in the objective function of the above linear program.

Objective 4: Minimizing c^l_{ij} given c^u_{ij} is minimum

Consider the linear programming formulation of the transportation problem. Let u_i and v_i be the dual variables corresponding to constraints (3.31) and (3.32). Alternative solution exist whenever

$$u_l + v_k = c_{lk} \tag{3.33}$$

for some l and k. To solve for Objective 4 we solve Objective 2 first. Next we identify the basic variables, as well as the non basic variables where some of the corresponding dual variables satisfy (3.33). Next we solve a TP with cost values given by the lower limit at the cells identified above. The remaining non basic variables will have $c_{ij} = \infty$.

LP formulation for Objective 5 of the cost parameter in TP

To attain the Objective 5 for transportation problem, we solve the system given below. The conditions on the selection of the desired interval $[L_d, U_d]$ are given in Section 2.7.

$$\begin{aligned}
\sum_{j=1}^{n} x_{ij} &\leq a_i \quad 1 \leq i \leq m \\
\sum_{i=1}^{m} x_{ij} &\geq b_i \quad 1 \leq j \leq n \\
\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}^{l} x_{ij} &\geq L_d \\
\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}^{u} x_{ij} &\leq U_d \\
x_{ij} &\geq 0
\end{aligned}$$

Example 3.5.1

Consider a problem with five distribution centers and five dealers with balanced supplies and demands. The data for this example is given in Figure 3.19. The interval values given in the figure are costs associated with the transportation of each unit from the distribution centre to the dealers. We solve this problem for all the objectives of the decision maker using linear programming.

Distribution Centers \ Dealers	1	2	3	4	5	Supplies
1	[10,11]	[12,14]	[16,20]	[8,13]	[7,15]	10
2	[6,9]	[8,14]	[7,11]	[9,16]	[8,13]	12
3	[5,7]	[8,10]	[14,17]	[7,16]	[10,14]	15
4	[12,16]	[7,12]	[11,16]	[17,21]	[18,23]	11
5	[7,12]	[12,15]	[7,14]	[10,13]	[8,16]	17
Demand	13	17	11	15	9	

Figure 3.19: Data for Example 3.5.1

The following is the general linear program to solve for the first three objectives. The objective function **Z** of this linear program changes for each objective but the constraints remain same.

$$Minimize\ \mathbf{Z}$$

Subject to

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 10$$
$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 12$$
$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 15$$
$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 11$$
$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} = 17$$
$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 13$$

$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 17$$
$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 11$$
$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 15$$
$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 9$$
$$x_{ij} \geq 0$$

Solving for Objective 1

Objective 1 is achieved by taking **Z** as follows.

$$\mathbf{Z} = 10x_{11} + 12x_{12} + 16x_{13} + 8x_{14} + 7x_{15} + 6x_{21} + 8x_{22} + 7x_{23} +$$
$$9x_{24} + 8x_{25} + 5x_{31} + 8x_{32} + 14x_{33} + 7x_{34} + 10x_{35} + 12x_{41} +$$
$$7x_{42} + 11x_{43} + 17x_{44} + 18x_{45} + 7x_{51} + 12x_{52} + 7x_{53} + 10x_{54} + 8x_{55}$$

The solution obtained for Objective 1 is shown in Figure 3.20 and the total cost

Distribution Centers	Dealers 1	2	3	4	5	Supplies
1	[10,11]	[12,14]	[16,20]	[8,13] 1	[7,15] 9	10
2	[6,9] 6	[8,14] 6	[7,11]	[9,16]	[8,13]	12
3	[5,7] 1	[8,10]	[14,17]	[7,16] 14	[10,14]	15
4	[12,16]	[7,12] 11	[11,16]	[17,21]	[18,23]	11
5	[7,12] 6	[12,15]	[7,14] 11	[10,13]	[8,16]	17
Demand	13	17	11	15	9	

Figure 3.20: Solution for Objective 1 of Example 3.5.1

is found to be [454,875]. The values obtained for for x_{ij}'s are given below the cost intervals.

Solving for Objective 2

Objective 2 is achieved by taking **Z** as follows.

$$\mathbf{Z} = 11x_{11} + 14x_{12} + 20x_{13} + 13x_{14} + 15x_{15} + 9x_{21} + 14x_{22} + 11x_{23} +$$
$$16x_{24} + 13x_{25} + 7x_{31} + 10x_{32} + 17x_{33} + 16x_{34} + 14x_{35} + 16x_{41} +$$
$$12x_{42} + 16x_{43} + 21x_{44} + 23x_{45} + 12x_{51} + 15x_{52} + 14x_{53} + 13x_{54} + 16x_{55}$$

The solution obtained for Objective 2 is shown in Figure 3.21 and the total cost

Dealers

Distribution Centers	1	2	3	4	5	Supplies
1	[10,11]	[12,14] 4	[16,20]	[8,13]	[7,15] 6	10
2	[6,9]	[8,14]	[7,11] 11	[9,16]	[8,13] 1	12
3	[5,7] 13	[8,10] 2	[14,17]	[7,16]	[10,14]	15
4	[12,16]	[7,12] 11	[11,16]	[17,21]	[18,23]	11
5	[7,12]	[12,15]	[7,14]	[10,13] 15	[8,16] 2	17
Demand	13	17	11	15	9	

Figure 3.21: Solution for Objective 2 of Example 3.5.1

is found to be [499,750]. The values obtained for for x_{ij}'s are given below the cost intervals.

Solving for Objective 3

The above linear program can be used to obtain Objective 3 which is to minimize the width of the total cost by taking **Z** by the follows.

$$\mathbf{Z} = x_{11} + 2x_{12} + 4x_{13} + 5x_{14} + 8x_{15} + 3x_{21} + 6x_{22} + 4x_{23} + 7x_{24} + 5x_{25} + 2x_{31} + 2x_{32} + 3x_{33} + 9x_{34} + 4x_{35} + 4x_{41} + 5x_{42} + 5x_{43} + 4x_{44} + 5x_{45} + 5x_{51} + 3x_{52} + 7x_{53} + 3x_{54} + 8x_{55}$$

The solution obtained for Objective 3 is shown in Figure 3.22 and the total cost

Dealers

Distribution Centers	1	2	3	4	5	Supplies
1	[10,11] 10	[12,14]	[16,20]	[8,13]	[7,15]	10
2	[6,9] 3	[8,14]	[7,11] 9	[9,16]	[8,13]	12
3	[5,7]	[8,10] 15	[14,17]	[7,16]	[10,14]	15
4	[12,16]	[7,12]	[11,16] 2	[17,21]	[18,23] 9	11
5	[7,12]	[12,15] 2	[7,14]	[10,13] 15	[8,16]	17
Demand	13	17	11	15	9	

Figure 3.22: Solution for Objective 3 of Example 3.5.1

is found to be [659,850]. The values obtained for for x_{ij}'s are given below the cost intervals.

Solving for Objective 4

Consider the solution obtained for Objective 2 given in Figure 3.21. The basic variables are $x_{12}, x_{15}, x_{23}, x_{25}, x_{31}, x_{32}, x_{42}, x_{54}$ and x_{55}. The non-basic variables where the sum of corresponding dual variables satisfy (3.33) are $x_{11}, x_{21}, x_{51}, x_{52}$ and x_{53}(to be added). The lower limits of the cost values are given in the cells identified for the basic and non-basic variables. The rest of the cells are given values ∞ as shown in Figure 3.23.

	Dealers 1	2	3	4	5	Supplies
Distribution Centers 1	10	12 **4**	∞	∞	7 **6**	10
2	6	∞	7 **11**	∞	8 **1**	12
3	5 **13**	8 **2**	∞	∞	∞	15
4	∞	7 **11**	∞	∞	∞	11
5	7	12	7	10 **15**	8 **2**	17
Demand	13	17	11	15	9	

Figure 3.23: Intermediate Tableau for Objective 4

The resulting tableau is now solved to obtain the solution for Objective 4. The solution obtained for Objective 4 is shown in Figure 3.24 and the total cost is found to be [487,750].

		Dealers					
		1	2	3	4	5	Supplies
Distribution Centers	1	[10,11]	[12,14] 1	[16,20]	[8,13]	[7,15] 9	10
	2	[6,9] 3	[8,14]	[7,11] 9	[9,16]	[8,13]	12
	3	[5,7] 10	[8,10] 5	[14,17]	[7,16]	[10,14]	15
	4	[12,16]	[7,12] 11	[11,16]	[17,21]	[18,23]	11
	5	[7,12]	[12,15]	[7,14] 2	[10,13] 15	[8,16]	17
Demand		13	17	11	15	9	

Figure 3.24: Solution for Objective 4

Solving for Objective 5

Objective 5 is to obtain a solution in a desired interval. Let the desired interval for our example be [500,800]. We have to solve the following system to achieve Objective 5.

$$10x_{11} + 12x_{12} + 16x_{13} + 8x_{14} + 7x_{15} + 6x_{21} + 8x_{22} + 7x_{23}+$$
$$9x_{24} + 8x_{25} + 5x_{31} + 8x_{32} + 14x_{33} + 7x_{34} + 10x_{35} + 12x_{41}+$$
$$7x_{42} + 11x_{43} + 17x_{44} + 18x_{45} + 7x_{51} + 12x_{52} + 7x_{53} + 10x_{54} + 8x_{55} \geq 500$$
$$11x_{11} + 14x_{12} + 20x_{13} + 13x_{14} + 15x_{15} + 9x_{21} + 14x_{22} + 11x_{23}+$$
$$16x_{24} + 13x_{25} + 7x_{31} + 10x_{32} + 17x_{33} + 16x_{34} + 14x_{35} + 16x_{41}+$$
$$12x_{42} + 16x_{43} + 21x_{44} + 23x_{45} + 12x_{51} + 15x_{52} + 14x_{53} + 13x_{54} + 16x_{55} \leq 800$$
$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 10$$
$$x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 12$$
$$x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 15$$

$$x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 11$$
$$x_{51} + x_{52} + x_{53} + x_{54} + x_{55} = 17$$
$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 13$$
$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 17$$
$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 11$$
$$x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 15$$
$$x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 9$$
$$x_{ij} \geq 0$$

The solution obtained for Objective 5 is shown in Figure 3.25 and the total cost is found to be [500,750] which is within the desired interval. The values obtained for for x_{ij}'s are given below the cost intervals.

Distribution Centers	Dealers 1	2	3	4	5	Supplies
1	[10,11] 3.4	[12,14]	[16,20]	[8,13]	[7,15] 6.6	10
2	[6,9] 0.6	[8,14]	[7,11] 11	[9,16]	[8,13] 0.4	12
3	[5,7] 9	[8,10] 6	[14,17]	[7,16]	[10,14]	15
4	[12,16]	[7,12] 11	[11,16]	[17,21]	[18,23]	11
5	[7,12]	[12,15]	[7,14]	[10,13] 15	[8,16] 2	17
Demand	13	17	11	15	9	

Figure 3.25: Solution for Objective 5 of Example 3.5.1

3.6 Assignment Problem (AP)

Definition

An important special case of the transportation problem is the case where $m = n$ and each $a_i = 1$ and each $b_j = 1$. This special case is called the *assignment problem.* To develop the integer program for this problem we can modify the transportation problem by assigning unit values for supplies and demands.

An efficient method known as the *Hungarian method* is used to solve the assignment problem. In this method the cost matrix c'_{ij}(matrix formed by putting the cost of assigning a job to a machine in a matrix form) is altered by subtracting constants from their original costs in order to give zero entries. If these zero elements or a subset thereof constitute a feasible solution, this solution is optimal, because cost cannot be negative.

The hungarian method used in the case of point cost values is given in Algorithm 3.4.

Objectives 1, 2 and 3

The first three objectives can be achieved by applying the Hungarian method(explained above). To achieve Objective 1 we have to consider only the lower limits of the cost interval and apply Hungarian method. If we use only upper limits of the cost interval and apply Hungarian method we will achieve Objective 2. To achieve Objective 3 we

Step 1: Subtract the smallest element in each row(column) from the corresponding row(column) of the cost matrix.
Step 2: Check if each of the rows and columns of the cost matrix have zero element(s). If yes go to step 4, else go to step 3.
Step 3: Select the row(or column) which has all non-zero elements and subtract the smallest of elements of this row(or column) from each of the elements of this row(or column)
Step 4: Check if the zero elements constitute an feasible assignment. If yes Stop, else go to step 5.
Step 5: Draw minimum number of lines through some of the rows and columns such that all the zeros are crossed out.
Step 6: Select smallest uncrossed out element and add it to every element at the intersection of two lines and subtract it from every uncrossed out element. Go to step 4.

Algorithm 3.4: Algorithm for Hungarian method

have to use the widths (refer Section 2.5) of the cost intervals and apply hungarian method. The solution obtained will give the minimum width in the total cost.

Objective 4: Minimizing c_{ij}^l given c_{ij}^u is minimum

If there are alternative solutions for Objective 2 then we can search among these solutions for the solution of Objective 4. The procedure is as follows. Consider the final table of the assignment problem obtained when solving Objective 2. Identify the cells which has zero values. Next construct the assignment table when the cost in the identified cell is equal to the lower limit of the cost. Use ∞ as the cost in the other cells. The solution of the resulting assignment problem satisfies objective 4.

Example 3.6.1

Here we are considering a four job four machine assignment problem. The data for this example is given in Figure 3.26(a). By applying the Hungarian method for the upper limit values of the cost intervals we reach from Figures 3.26(b) to 3.26(e). The final Figure 3.26(e) gives the feasible solutions for this example.

The assignment problem to solve next to achieve the solution of Objective 4 is shown in Figure 3.27.The final solution is shown in Figure 3.28.The optimal solution is to assign job 1 to machine 2, 2 to 3, 3 to 1 and 4 to 4. This assignment costs [15,25].

Objective 5: Obtaining a solution in a desired interval

To solve the assignment problem for Objective 5 we introduce the integer program for the point cost parameter case. Let c_{ij} be the cost of assigning a job i to machine j and let

$$x_{ij} = \begin{cases} 1 & \text{if job } i \text{ is assigned to machine } j \\ 0 & \text{otherwise} \end{cases}$$

The integer program will thus be of the form given below.

$$Minimize \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad 1 \leq i \leq n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad 1 \leq j \leq n$$

	1	2	3	4
1	(3,5)	(4,7)	(6,10)	(5,6)
2	(8,11)	(6,8)	(4,7)	(5,9)
3	(2,4)	(3,7)	(3,6)	(4,5)
4	(9,12)	(6,9)	(4,7)	(5,7)

(a)

	1	2	3	4
1	5	7	10	6
2	11	8	7	9
3	4	7	6	5
4	12	9	7	7

(b)

	1	2	3	4
1	0	2	5	1
2	4	1	0	2
3	0	3	2	1
4	5	2	0	0

(c)

	1	2	3	4
1	0	1	5	1
2	4	0	0	2
3	0	2	2	1
4	5	1	0	0

(d)

	1	2	3	4
1	0	0	4	0
2	5	0	0	2
3	0	1	1	0
4	6	1	0	0

(e)

Figure 3.26: Example of Assignment Problem for Objective 4.

	1	2	3	4
1	3	4	∞	5
2	∞	6	4	∞
3	2	∞	∞	4
4	∞	∞	4	5

Figure 3.27: Breaking ties to minimize lower limit.

	1	2	3	4
1	0	0	∞	1
2	∞	1	0	∞
3	0	∞	∞	1
4	∞	∞	0	0

Figure 3.28: Optimal Solution for Objective 4.

$$x_{ij} = 0 \text{ or } 1$$

To obtain the Objective 5 for assignment problem, we have to solve the system given below. $[c_{ij}^l, c_{ij}^u]$ is the interval of the cost of assigning a job i on machine j. The conditions for the decision maker to choose the desired interval $[L_d, U_d]$ in which he has to select the limits are given in Section 2.7.

$$\sum_{j=1}^{n} x_{ij} = 1 \quad 1 \le i \le n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad 1 \le j \le n$$

$$\sum_i \sum_j c_{ij}^l x_{ij} \ge L_d$$

$$\sum_i \sum_j c_{ij}^u x_{ij} \le U_d$$

$$x_{ij} = 0 \text{ or } 1$$

Example 3.6.2

Consider the data given in Figure 3.26 (a) for the four job four machine problem. Objective 5 is to obtain a solution in a desired interval say [19,29] for this problem. The system developed for the data of this example is as follows.

$$3x_{11} + 4x_{12} + 6x_{13} + 5x_{14} + 8x_{21} + 6x_{22} + 4x_{23} + 5x_{24} +$$
$$2x_{31} + 3x_{32} + 3x_{33} + 4x_{34} + 9x_{41} + 6x_{42} + 4x_{43} + 5x_{44} \le 19$$
$$5x_{11} + 7x_{12} + 10x_{13} + 6x_{14} + 11x_{21} + 8x_{22} + 7x_{23} + 9x_{24} +$$
$$4x_{31} + 7x_{32} + 6x_{33} + 5x_{34} + 12x_{41} + 9x_{42} + 7x_{43} + 7x_{44} \le 29$$
$$x_{11} + x_{12} + x_{13} + x_{14} = 1$$
$$x_{21} + x_{22} + x_{23} + x_{24} = 1$$
$$x_{31} + x_{32} + x_{33} + x_{34} = 1$$

$$x_{41} + x_{42} + x_{43} + x_{44} = 1$$
$$x_{11} + x_{21} + x_{31} + x_{41} = 1$$
$$x_{12} + x_{22} + x_{32} + x_{42} = 1$$
$$x_{13} + x_{23} + x_{33} + x_{43} = 1$$
$$x_{14} + x_{24} + x_{34} + x_{44} = 1$$
$$x_{ij} = 0 \text{ or } 1$$

The solution obtained for the above system is the assignment of job 1 to machine 3, 2 to 2, 3 to 1, and 4 to 4 and the cost involved in this assignment is [19,29].

Chapter 4

Scheduling Problems

4.1 Introduction

In this chapter we consider some single machine scheduling problems. The following four assumptions are valid throughout this chapter. These assumptions are basically extended version of those for general scheduling problems [49]. Here, instead of point values we use interval values for processing times which is the base of this study.

1. *Each job is an entity*, which means that no two operations can be processed simultaneously on a single job.

2. *Pre-emption* is not allowed.

3. All the jobs are ready at time zero.

4. No idle times are allowed i.e., a job is started once the machine becomes free.

The following definitions and notation will be used through out this chapter. The superscript l is used to denote the lower limit and the superscript u for the upper limit for the respective parameter.

n = number of jobs

J_i = is the ith job, $1 \leq i \leq n$.

$[p_i^l, p_i^u]$ = is the processing time interval of job J_i.

d_i = is the due date, i.e. the promised delivery date of job J_i

$[C_i^l, C_i^u]$ = is the completion time interval of J_i. This is the earliest time and the latest time at which the processing of J_i finishes. If J_i is in position k then $[C_i^l, C_i^u] = [\sum_{j=1}^{k} p_{(j)}^l, \sum_{j=1}^{k} p_{(j)}^u]$, where $[p_{(j)}^l, p_{(j)}^u]$ is the processing time of the jth job in the sequence.

$[L_i^l, L_i^u]$ = is the lateness interval of J_i. This is the difference between its completion time and its due date. Thus $L_i^l = C_i^l - d_i$ and $L_i^u = C_i^u - d_i$

$[T_i^l, T_i^u]$ = is the tardiness interval of J_i. A job is tardy when it completes after its due date. Thus $[T_i^l, T_i^u] = [max(C_i^l - d_i, 0), max(C_i^u - d_i, 0)]$.

In this chapter we introduce a new objective of obtaining a robust schedule (Objective 6) in addition to the five objectives, viz., minimizing the lower limit of an objective function, minimizing the upper limit, minimizing the width, minimizing the lower limit given the upper limit is at its minimum and obtaining a solution in a desired interval. In the next section we define the concept of a robust schedule [2] and describe how to obtain it. We adopt this objective function in some scheduling problems to illustrate it.

In Section 4.3 we consider the mean flow time problem. We prove that Objective 3, minimizing the width of flow time, can be achieved if the jobs are arranged in increasing order of their respective width $(p_i^u - p_i^l)$. It is shown that Objective 4 can be solved after solving Objective 2. Among the alternative solutions for Objective 2 we search for the solution of Objective 4 by breaking ties with respect to lower limits of the processing times. Objectives 5 is realized by formulating the problem as an integer program. Finally we report our empirical results for obtaining a robust schedule.

In Section 4.4, minimizing maximum lateness problem, will be considered and solved for all objectives given in Section 2.4. Section 4.5 deals with the problem of minimizing the number tardy jobs. In Section 4.6 minimizing the sum of earliness and tardiness problem is solved for the interval processing times case. A new method to solve this problem is proposed and we prove that it gives the optimal solution. In Section 4.7 the problem of minimizing the weighted sum of earliness and tardiness for jobs with different weights will be solved for interval processing times. A new method to solve this problem is proposed and its optimality is proved.

4.2 Robust Schedule

In this chapter a new objective is introduced which is to find a robust schedule for some performance measures in an interval case. A *robust schedule* can be defined as a schedule which has the minimum worst case performance.

This concept of a robust schedule has been introduced by Bintong et.al., [2]. They used this concept in project management with interval durations. In the following we describe the approach for generating a robust schedule [2]. We consider all possible schedules of the jobs and for each schedule we compute the worst case performance. A robust schedule is one which has the least worst case performance. To compute the worst case performance of a schedule we generate all possible combinations of job processing times. Each combination is called a *scenario.*

Let $\mathbf{p} = (p_1, p_2, .., p_n)$ be some combination of job processing times. Let $f_s(\mathbf{p})$ be the performance measure corresponding to a given sequence of job, S, and processing times $\mathbf{p}$. Given $\mathbf{p}$, find the optimal sequence. Let $f^*(\mathbf{p})$ be the corresponding value of the performance measure. The penalty of schedule S is,

$$P_S = max_{\mathbf{p}}\{f_s(\mathbf{p}) - f^*(\mathbf{p})\}$$

4.3 The Mean Flow Time Problem ($\overline{F}$)

The flow time is defined as the time spent by the job in the work shop. The objective is to minimize, the mean flow time. Since the processing times are intervals, the mean flow time will be an interval, $\overline{F} = [\overline{F}^l, \overline{F}^u]$, where $\overline{F}^l$ is the lower limit and $\overline{F}^u$ is the upper limit of the mean flow time interval. We consider the above problem for different objectives given in Section 2.4.

The shortest processing time (SPT) rule gives the optimal solution for the mean flow problem with point processing times. This has been proved by many authors in different ways. Conway, Maxwell and Miller [7] had developed a proof in 1967 based on the area under a concave graph. Rinnooy Kan [46] had provided an algebraic proof in 1976. Other authors like French [49], Baker [27] also have given simple proofs based on a pairwise interchange technique. We extend this SPT rule to solve the mean flow time problem with interval processing times for different options of the decision maker.

Objectives 1 and 2: Minimizing $\overline{F}^l$ / $\overline{F}^u$

Our objective is to obtain an optimal solution by minimizing $\overline{F}^l (or \overline{F}^u)$. In this case the problem is reduced to mean flow time problem with point processing times. To achieve Objective 1 we consider only the lower limits of the processing times and obtain the optimal solution using the shortest processing time (SPT) rule for the lower limits only. Objective 2 can be achieved by doing the same for the upper limits of processing times.

Objective 3 : Minimizing $\overline{F}^u - \overline{F}^l$

In the following theorem we show that the Objective 3 i.e., minimizing the width of the mean flow time interval is accomplished by arranging the jobs in ascending order of their respective widths $w_i = (p_i^u - p_i^l)$.

Theorem 4.3.1:

For a single machine scheduling problem with interval processing times, the width of the mean flow time interval, $(\overline{F}^u - \overline{F}^l)$, is minimized by sequencing jobs such that

$$w_{(1)} \leq w_{(2)} \leq \ldots . \leq w_{(n)}$$

where $w_{(k)} = p_{(k)}^u - p_{(k)}^l$ is the width of the processing time intervals of the kth job in the sequence, $J_{(k)}$.

Proof:

Since only permutation schedules need be considered, the flow-time of the job in the kth position of an arbitrary sequence is simply

$$F_{(k)} = \sum_{i=1}^{k} p_{(i)} = \sum_{i=1}^{k} [p_{(i)}^l, p_{(i)}^u]$$

$$\begin{aligned}
\overline{F} &= \frac{1}{n} \sum_{k=1}^{n} F_{(k)} = \frac{1}{n} \sum_{k=1}^{n} \sum_{i=1}^{k} p_{(i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} (n - i + 1) p_{(i)} \\
&= \frac{1}{n} [\sum_{i=1}^{n} (n - i + 1) p_{(i)}^l, \sum_{i=1}^{n} (n - i + 1) p_{(i)}^u]
\end{aligned}$$

The width of $\overline{F}$ is given by

$$W_{\overline{F}} = \frac{1}{n}\sum_{i=1}^{n}(n-i+1)(p^u_{(i)} - p^l_{(i)})$$
$$= \frac{1}{n}\sum_{i=1}^{n}(n-i+1)w_{(i)}$$

A sum of pairwise products of two sequences of numbers will be minimized if one sequence is arranged in increasing order and the other in decreasing order [19]. Since, the numbers $(n-i+1)$ are already in decreasing order, the minimization is accomplished by sequencing the jobs so that the widths of the processing times are in a non-decreasing order. This completes the proof.

Objective 4 : Minimizing $\overline{F}^l$ given $\overline{F}^u$ is minimum.

Objective 4 is achieved by applying the SPT rule for the upper limit of the processing times (p_i^u's) and in case of ties we apply SPT for the lower limits of processing times (p_i^l's) for this subset of jobs. In case there are no ties then we will get the same schedule as obtained for Objective 2.

Example 4.3.1

Consider a six job problem. The processing time for each job are given in Table 4.1. Now we apply the SPT rule with respect to the upper limits of the processing times. Since the jobs 3, 5 and 6 have the same upper limit of their processing times, we have six alternative schedules. Table 4.2 gives the six schedules along with the mean flow time calculated for each one of them. The solution for Objective 4 is the

Table 4.1: Data for Example 4.3.1

Jobs	1	2	3	4	5	6
Processing Times	[1,2]	[4,5]	[5,8]	[7,10]	[3,8]	[7,8]

Table 4.2: Alternative Schedules and Mean Flow Time

S.No	Schedules	Mean Flow Time
1	$J_1 - J_2 - J_3 - J_5 - J_6 - J_4$	$[12.67, 19.83]$
2	$J_1 - J_2 - J_3 - J_6 - J_5 - J_4$	$[13.33, 19.83]$
3	$J_1 - J_2 - J_6 - J_3 - J_5 - J_4$	$[13.67, 19.83]$
4	$J_1 - J_2 - J_6 - J_5 - J_3 - J_4$	$[13.33, 19.83]$
5	$J_1 - J_2 - J_5 - J_6 - J_3 - J_4$	$[12.67, 19.83]$
6	$J_1 - J_2 - J_5 - J_3 - J_6 - J_4$	$[12.33, 19.83]$

one with the least lower limit for the mean flow time interval. This solution can be directly obtained by applying the SPT rule to the lower limits of the processing times for the jobs which have same upper limit.

Hence, when SPT rule is applied to the lower limits of processing times for jobs 3, 5 and 6, the sequence within these jobs will now be $J_5 - J_3 - J_6$. The optimal sequence for this example will thus be $J_1 - J_2 - J_5 - J_3 - J_6 - J_4$, with mean flow time of $[12.33, 19.83]$.

Objective 5: To get a solution within a desired interval

Unlike the case of the first four objectives of the decision maker for the mean flow time problem, Objective 5, which is to obtain a solution within a desired interval, cannot be solved using SPT technique. We therefore propose the linear programming appraoch in order to solve the mean flow time problem for Objective 5. First consider the case where the processing times are point values.

Let $X_{ij} = \begin{cases} 1 & \text{if job } i \text{ is at } j\text{th position in the sequence} \\ 0 & \text{otherwise} \end{cases}$

Then, we must have

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad 1 \leq j \leq n \tag{4.1}$$

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad 1 \leq i \leq n \tag{4.2}$$

Let p_i be the processing time of job i, and C_j be the completion time of any job in position j. Let us consider the first position in the sequence. Since any job can be placed in this position the completion time at this position will be given as

$C_1 = X_{11}p_1 + X_{21}p_2 + X_{31}p_3 + .. + X_{n1}p_n$

Similarly if we consider position two, a job i can be placed in either of the two positions and hence we can write the completion time at this position and so on as

$C_2 = (X_{11} + X_{12})p_1 + (X_{21} + X_{22})p_2 + (X_{31} + X_{32})p_3 + .. + (X_{n1} + X_{n2})p_n$

.

.

.

$C_n = (X_{11} + X_{12} + .. + X_{1n})p_1 + (X_{21} + X_{22} + .. + X_{2n})p_2 + (X_{31} + X_{32} + .. + X_{3n})p_3 +$

$\quad .. + (X_{n1} + X_{n2} + .. + X_{nn})p_n$

$\quad = p_1 + p_2 + p_3 + .. + p_n$

Let F be the total flow time, that is the sum of all the completion times, calculated as

$$F = C_1 + C_2 + .. + C_n$$

$$= (X_{11})p_1 + (X_{21})p_2 + (X_{31})p_3 + .. + (X_{n1})p_n + (X_{11} + X_{12})p_1 + (X_{21} + X_{22})p_2$$

$$+(X_{31} + X_{32})p_3 + .. + (X_{n1} + X_{n2})p_n + .. + p_1 + p_2 + p_3 + .. + p_n$$

After rearranging terms we get

$$F = p_1[X_{11} + (X_{11} + X_{12}) + (X_{11} + X_{12} + X_{13}) + . + 1]$$

$$+p_2[X_{21} + (X_{21} + X_{22}) + (X_{21} + X_{22} + X_{23}) + . + 1]$$

$$.. + p_n[X_{n1} + (X_{n1} + X_{n2}) + (X_{n1} + X_{n2} + X_{n3}) + 1]$$

$$= p_1[(n-1)X_{11} + (n-2)X_{12} + (n-3)X_{13} + ... + 1]$$

$$+p_2[(n-1)X_{21} + (n-2)X_{22} + (n-3)X_{23} + ... + 1]$$

$$.. + p_n[(n-1)X_{n1} + (n-2)X_{n2} + (n-3)X_{n3} + ... + 1]$$

$$= p_1[\sum_{i=1}^{n-1}(n-i)X_{1i} + 1] + p_2[\sum_{i=1}^{n-1}(n-i)X_{2i} + 1] + .. + p_n[\sum_{i=1}^{n-1}(n-i)X_{ni} + 1]$$

Hence the total flow time of all the jobs can be written in a general form as

$$F = \sum_{k=1}^{n} p_k[\sum_{i=1}^{n-1}(n-i)X_{ki}+1]$$

Now we introduce the interval processing times in the above expression. Let $p_i = [p_i^l, p_i^u]$ be the processing time of job i. Let the total flow time fall in an interval $F = [F^l, F^u]$, then

$$F^l = \sum_{k=1}^{n} p_k^l[\sum_{i=1}^{n-1}(n-i)X_{ki} + 1] \tag{4.3}$$

$$F^u = \sum_{k=1}^{n} p_k^u [\sum_{i=1}^{n-1} (n-i) X_{ki} + 1] \quad (4.4)$$

The mean flow time is obtained from Equations (4.3) and (4.4). Let $\overline{F} = [\overline{F}^l, \overline{F}^u]$ be the mean flow time then

$$\overline{F}^l = \frac{1}{n} \sum_{k=1}^{n} p_k^l [\sum_{i=1}^{n-1} (n-i) X_{ki} + 1] \quad (4.5)$$

$$\overline{F}^u = \frac{1}{n} \sum_{k=1}^{n} p_k^u [\sum_{i=1}^{n-1} (n-i) X_{ki} + 1] \quad (4.6)$$

Equations (4.1), (4.2), (4.5) and (4.6) can now be solved for the binary variable X_{ij} to obtain the mean flow time.

The conditions to select the desired limits for Objective 5 of the decision maker are given in Section 2.7. Let $[\overline{F}_d^l, \overline{F}_d^u]$ be the interval within which the mean flow time is desired. Two constraints $\overline{F}^l \geq \overline{F}_d^l$ and $\overline{F}^u \leq \overline{F}_d^u$ should be added to Equations (4.1), (4.2), (4.5) and (4.6) to form the following system to solve for Objective 5.

$$\begin{aligned}
\overline{F}^l &= \frac{1}{n} \sum_{k=1}^{n} p_k^l [\{\sum_{i=1}^{n-1} (n-i) X_{ki}\} + 1] \\
\overline{F}^u &= \frac{1}{n} \sum_{k=1}^{n} p_k^u [\{\sum_{i=1}^{n-1} (n-i) X_{ki}\} + 1] \\
& \sum_{j=1}^{n} X_{ij} = 1 \qquad\qquad 1 \leq i \leq n \\
& \sum_{i=1}^{n} X_{ij} = 1 \qquad\qquad 1 \leq j \leq n \\
& \overline{F}^l \geq \overline{F}_d^l
\end{aligned}$$

$$\overline{F}^u \leq \overline{F}^u_d$$

$$X_{ij} = 0 \; or \; 1$$

Example 4.3.2

Consider a 6 job single machine problem with the processing times as given in Table 4.3. The objective of this problem to achieve a sequence which gives the mean flow time in the interval [30,40]. The 0-1 program developed based on the data is as follows.

Table 4.3: Data for Example 4.3.2

Jobs	1	2	3	4	5	6
Processing Times	[12,15]	[7,8]	[9,14]	[5,7]	[15,19]	[10,16]

$$6\overline{F}^u = 75X_{11} + 60X_{12} + 45X_{13} + 30X_{14} + 15X_{15} + 40X_{21} + 32X_{22}$$
$$+24X_{23} + 16X_{24} + 8X_{25} + 70X_{31} + 56X_{32} + 42X_{33} + 28X_{34} + 14X_{35}$$
$$+35X_{41} + 284X_{42} + 21X_{43} + 14X_{44} + 7X_{45} + 95X_{51} + 76X_{52} + 57X_{53}$$
$$+38X_{54} + 19X_{55} + 80X_{61} + 64X_{62} + 48X_{63} + 32X_{64} + 16X_{65} + 79$$
$$6\overline{F}^l = 60X_{11} + 48X_{12} + 36X_{13} + 24X_{14} + 12X_{15} + 35X_{21} + 28X_{22}$$
$$+21X_{23} + 14X_{24} + 7X_{25} + 45X_{31} + 36X_{32} + 27X_{33} + 18X_{34} + 9X_{35}$$
$$+25X_{41} + 20X_{42} + 15X_{43} + 10X_{44} + 5X_{45} + 75X_{51} + 60X_{52} + 45X_{53}$$
$$+30X_{54} + 15X_{55} + 50X_{61} + 40X_{62} + 30X_{63} + 20X_{64} + 10X_{65} + 58$$
$$X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} = 1$$
$$X_{21} + X_{22} + X_{23} + X_{24} + X_{25} + X_{26} = 1$$
$$X_{31} + X_{32} + X_{33} + X_{34} + X_{35} + X_{36} = 1$$
$$X_{41} + X_{42} + X_{43} + X_{44} + X_{45} + X_{46} = 1$$
$$X_{51} + X_{52} + X_{53} + X_{54} + X_{55} + X_{56} = 1$$

$$X_{61} + X_{62} + X_{63} + X_{64} + X_{65} + X_{66} = 1$$
$$X_{11} + X_{21} + X_{31} + X_{41} + X_{51} + X_{61} = 1$$
$$X_{12} + X_{22} + X_{32} + X_{42} + X_{52} + X_{62} = 1$$
$$X_{13} + X_{23} + X_{33} + X_{43} + X_{53} + X_{63} = 1$$
$$X_{14} + X_{24} + X_{34} + X_{44} + X_{54} + X_{64} = 1$$
$$X_{15} + X_{25} + X_{35} + X_{45} + X_{55} + X_{65} = 1$$
$$X_{16} + X_{26} + X_{36} + X_{46} + X_{56} + X_{66} = 1$$
$$\overline{F}^l \geq 30$$
$$\overline{F}^u \leq 40$$
$$X_{ij} = 0 \text{ } or \text{ } 1$$

The schedule obtained from this 0-1 program is $J_4 - J_2 - J_1 - J_3 - J_5 - J_6$ with mean flow time of [30,39.67].

Objective 6: Robust Schedule for Mean Flow time

The method of obtaining a robust schedule is already explained in the beginning of this chapter. We prove in Theorem 4.3.2. that to find the worst performance of a schedule one has to consider only extreme values of intervals p_i. This is also valid for the mean flow time problem.

Let $J_1, J_2, ..J_n$ be a given sequence S and $p_1, p_2, .., p_n$ be some realizations of processing times. Then the total flow time $F = \sum_{i=1}^{n}(n - i + 1)p_i$. If the jobs are arranged according to the SPT rule then minimum total flow time $F_{SPT} = \sum_{i=1}^{n}(n - i + 1)p_{(i)}$ where $p_{(1)} \leq p_{(2)} \leq .. \leq p_{(n)}$ are the processing times.

Theorem 4.3.2:

The maximum of $\sum_{i=1}^{n}(n-i+1)p_i - \sum_{i=1}^{n}(n-i+1)p_{(i)}$ is achieved if $p_i = l_i$ *or* u_i

Proof:

Consider D given by;

$$\begin{aligned} D &= \sum_{i=1}^{n}(n-i+1)p_i - \sum_{i=1}^{n}(n-i+1)p_{(i)} \\ &= \sum_{i=1}^{n}(n-i+1)(p_i - p_{(i)}) \end{aligned} \tag{4.7}$$

Since the numbers p_i and $p_{(i)}$, $1 \le i \le n$ are the same values but in different sequence, then (4.7) can be written as

$D = \sum_{i=1}^{n} a_i p_i$ for some scalars a_i.

The maximum of D is achieved by choosing p_i as follows;

$$p_i = \begin{cases} u_i & if\ a_i \ge 0 \\ l_i & if\ a_i < 0 \end{cases}$$

This completes the proof.

A C program is developed to obtain a robust schedule for the mean flow time problem. In this code all possible sequences are generated. For each sequence all combinations of processing times were considered where $p_i = l_i$ *or* u_i The robust schedule is the schedule which yields minimum worst case performance (refer Section 4.2).

Example 4.3.3

The processing times for a 4 job problem are given in Table 4.4. The objective is to find a robust schedule for the mean flow time. The robust schedule obtained

Table 4.4: Data for Example 4.3.3

jobs	1	2	3	4
$[p^l, p^u]$	[3,6]	[8,10]	[5,12]	[6,7]

from the C program is $J_1 - J_4 - J_3 - J_2$. The worst case performance takes place for point processing times 6, 8, 12 and 7 for jobs 1, 2, 3 and 4 respectively. The mean flow time of the robust schedule using these point processing times is 19.25. For these processing times the optimal sequence is $J_1 - J_4 - J_2 - J_3$ which has a mean flow time of 18.25. Hence the worst case performance corresponding to sequence $J_1 - J_4 - J_3 - J_2$ is 19.25 - 18.25 = 1

4.4 Minimizing the Maximum Lateness Problem

Given assumption 4 in Section 4.1, we now study the problem of minimizing the maximum lateness. The lateness of a job i, $L_i = [L_i^l, L_i^u]$ is defined as the difference between its completion time $C_i = [C_i^l, C_i^u]$ and its due date d_i. Hence, lateness interval can be written as

$$[L_i^l, L_i^u] = [C_i^l - d_i, C_i^u - d_i] = [\sum_{j=1}^{i} p_{(j)}^l - d_i, \sum_{j=1}^{i} p_{(j)}^u - d_i]$$

Let $L_{max} = [L_{max}^l, L_{max}^u]$ be the maximum lateness interval where $L_{max}^l = max(L_1^l, L_2^l, ..L_n^l)$ and $L_{max}^u = max(L_1^u, L_2^u, ..L_n^u)$. Therefore L_{max} measures the worst violation of the due dates. We now form the problem of scheduling n jobs such that the maximum lateness $[L_{max}^l, L_{max}^u]$ is minimized.

The earliest due date (EDD) scheduling rule is known to give the optimal solution for the maximum lateness problem where the processing times are point values. The EDD rule is to schedule the jobs in the non-decreasing order of their due dates. This result is due to Jackson [23] who presented it in his research report in 1955.

Proposition 1:

The sequence S' produced by the earliest due date (EDD) rule will minimize any linear function of the form $aL^l_{max} + bL^u_{max}$.

Proof:

To minimize L^l_{max} we consider the set $p^l_i, 1 \leq i \leq n$. The optimal solution is to arrange the jobs in non-decreasing order of their due dates (EDD rule). Similarly, to minimize L^u_{max} we use the EDD rule. Since we are considering only the due dates in both the cases we obtain the same sequence. Considering any combination of the limits will also give the same EDD sequence. Hence, the minimum of $aL^l_{max} + bL^u_{max}$ is also an EDD sequence. This completes the proof.

It can be concluded that the schedule obtained by EDD sequencing will be the solution for the first five objectives of the decision maker (refer to Section 2.4).

Proposition 2:

The robust schedule for the maximum lateness problem is also an EDD sequence.

Proof:

We know that the EDD rule gives the optimal solution for the maximum lateness problem. Robust schedule is obtained by calculating the worst case performance for each sequence one of which is surely an EDD sequence. This sequence will definitely give zero penalty and hence will be selected as a robust schedule.

4.5 Minimizing number of Tardy Jobs

Based on the EDD sequence, Moore and Hodgson in 1968 developed an algorithm which gives the minimum number of tardy jobs. Sturm [51] gave an optimality proof for Moore's sequencing algorithm in 1970. Kise, et al., [29] extended this algorithm for the number of tardy jobs for non-zero ready times, which increase with the due dates of the respective jobs.

Since we are dealing with interval processing times we can have two approaches for solving this problem. In Approach I we consider either lower or upper limits of the processing times and in Approach II we consider the interval itself.

4.5.1 Approach I

Given a sequence of jobs we can find the minimum number of tardy jobs and the maximum number of tardy jobs by constructing the following sets.

$S_l = \{j \mid C_j^l > d\}$; and,

$S_u = \{j \mid C_j^u > d\}$

The minimum number of tardy jobsfor given sequence, $n_T^l = |S_l|$ and the maximum number of tardy jobs for the given sequence, $n_T^u = |S_u|$ where $[C_j^l, C_j^l]$ is the completion time of job j in the sequence. Moore's algorithm [49] is known to produce optimum solution for the minimum number of tardy jobs problem for point processing times.

4.5.1.1 Objectives 1 and 2

Objectives 1 is to minimize the lower limit of the number of tardy jobs. So we propose the use of Moore's algorithm using p_i^l. For the resulting sequence we construct the sets S_l and S_u defined above and the corresponding n_T^l will be the solution of Objective 1. For Objective 2 we use the sequence produced by Moore's algorithm where p_i^u are the processing times instead of p_i^l and the corresponding n_T^u will be the solution.

4.5.1.2 Objectives 3, 4 and 5

Objective 3 is to obtain minimum width of the interval $[n_T^l, n_T^u]$, Objective 4 is to minimize n_T^l, given n_T^u is minimum and Objective 5 is to obtain the number of tardy jobs in a desired interval. These objectives cannot be achieved by applying Moore's algorithm. Other approaches should hence be applied in order to achieve them. We propose integer programming approach to solve the problem for these objectives.

4.5.1.2.a Formulation for minimizing number of Tardy Jobs

Moore's Algorithm cannot be used to solve the tardy jobs problem for objectives 3, 4 and 5. Integer programs can be developed to solve the problem for each of these objectives. First, consider the case where the the processing times are given as point values p_i.

$$\text{Let } X_{ij} = \begin{cases} 1, & \text{if job } i \text{ is scheduled in position } j \\ 0 & \text{otherwise} \end{cases}$$

Then

$$\sum_{i=1}^{n} X_{ij} = 1 \qquad 1 \leq j \leq n \tag{4.8}$$

$$\sum_{j=1}^{n} X_{ij} = 1 \qquad 1 \leq i \leq n \tag{4.9}$$

Let C_{ij} be the completion time of job i in jth position in a given sequence. Then,

$C_{i1} = p_i$

$$C_{i2} = X_{11}p_i + X_{21}p_2 + ..X_{i-1,1}p_{i-1} + X_{i+1,1}p_{i+1} + .. + X_{n1}p_n = p_i + \sum_{k=1,k\neq i}^{n} X_{k1}p_k$$

.

.

$$C_{ij} = p_i + \sum_{k=1,k\neq i}^{n} p_k \sum_{l=1}^{j-1} X_{kl}$$

Since job i could be exactly in one position then the completion time of job i, will be given by

$$C_i = X_{i1}C_{i1} + X_{i2}C_{i2} + ... + X_{in}C_{in} = \textstyle\sum_{j=1}^{n} X_{ij}C_{ij} \qquad 1 \leq i \leq n$$

Now introduce the interval processing times $p_i = [p_i^l, p_i^u]$. Hence the completion time of job i in position j becomes $C_{ij} = [C_{ij}^l, C_{ij}^u]$. Where

$$C_{ij}^l = p_i^l + \sum_{k=1,k\neq i}^{n} p_k^l \sum_{l=1}^{j-1} X_{kl}$$

$$C_{ij}^u = p_i^u + \sum_{k=1,k\neq i}^{n} p_k^u \sum_{l=1}^{j-1} X_{kl}$$

For interval processing times we have $C_i = [C_i^l, C_i^u]$, where

$$C_i^l = \sum_{j=1}^{n} X_{ij} C_{ij}^l \qquad 1 \leq i \leq n \tag{4.10}$$

$$C_i^u = \sum_{j=1}^{n} X_{ij} C_{ij}^u \qquad 1 \leq i \leq n \tag{4.11}$$

The substitution of C_{ij}'s in (4.10) and (4.11) will lead to the cross products of the variables X_{ij}. Glover and Woolsey [17] suggested a procedure for problems having product of two integer variables. The procedure replaces the product, in our case $X_{ij}X_{kl}$ by a continous variable Z_{ijkl}, and introduces three constraints for each product as follows.

$$Z_{ijkl} \leq X_{ij} \tag{4.12}$$

$$Z_{ijkl} \leq X_{kl} \tag{4.13}$$

$$X_{ij} + X_{kl} - 1 \leq Z_{ijkl} \tag{4.14}$$

Let, TL_i and TU_i be two binary variables defined as follows,

$$TL_i = \begin{cases} 1, & \text{if } C_i^l > d_i \\ 0 & \text{otherwise} \end{cases}$$

$$TU_i = \begin{cases} 1, & \text{if } C_i^u > d_i \\ 0 & \text{otherwise} \end{cases}$$

The following constraints can be used to compute TL_i and TU_i. Let α be a large number and ϵ is a very small positive value. Then

$$\alpha\, TL_i \geq C_i^l - d_i \tag{4.15}$$

$$\alpha(1 - TL_i) \geq d_i - C_i^l + \epsilon \tag{4.16}$$

$$\alpha\, TU_i \geq C_i^u - d_i \tag{4.17}$$

$$\alpha(1 - TU_i) \geq d_i - C_i^u + \epsilon \tag{4.18}$$

If $C_i^l > d_i$ then from (4.15), $TL_i = 1$. On the other hand if $C_i^l \leq d_i$ then from (4.16) $TL_i = 0$. Similarly if $C_i^u > d_i$ then from (7), $TU_i = 1$ and if $C_i^u \leq d_i$ then from (8) $TU_i = 0$.

We propose to select α as follows. $\alpha = (\sum_{i=1}^{n} p_i^l) - d_i$ for constraints (4.15) and (4.16), and $\alpha = (\sum_{i=1}^{n} p_i^u) - d_i$ for constraints (4.17) and (4.18). If p_i and d_i are integers then ϵ can be taken less than 1.

4.5.1.2.b Objectives 3: Minimizing the width of the interval $[n_T^l, n_T^u]$

To solve the problem of minimizing the width of the number of tardy jobs interval $[n_T^l, n_T^u]$, the following integer program has to be solved.

$$Minimize \sum_{i=1}^{n} TU_i - \sum_{i=1}^{n} TL_i$$

subject to

Constraints (4.8) to (4.18)

$$X_{ij}, TL_i, TU_i \; = \; 0 \; or \; 1$$

This integer program will require large number of variables and constraints to solve the problem. For an n job problem this integer program will have $\frac{1}{2}[n(n-1)]^2 + n^2 + 4n$ variables and $\frac{3}{2}[n(n-1)]^2 + 8n$ constraints. For a 6 job problem there are 510 variables and 1398 constraints. We realize that this number of variables and constraints will take computationally large time. It will rather take less time to solve for all possible schedules to find the solution, but we propose this formulation for theoretical purpose of solving problems with different objectives of the decision maker.

Example 4.5.1.1

Consider a 6 job problem with processing times and due dates as given in Table 4.5.

Table 4.5: Data for Example 4.5.1.1

jobs	1	2	3	4	5	6
$[p^l, p^u]$	[12,15]	[7,8]	[9,14]	[5,7]	[15,19]	[10,16]
d_i	16	12	35	35	36	36

The integer program developed for this data is given in Appendix I, where the three constraints (a), (b) and (c) should be added for each of the variable Z_{ijkl} appearing in any constraint. The solution of this integer program gives the schedule

Table 4.6: Solution for Example 4.5.1.1

Schedule	5	2	1	6	4	3
Completion times	[15,19]	[22,27]	[34,42]	[44,58]	[49,65]	[58,79]
Due dates(d_i)	36	12	16	36	35	35

$J_5 - J_2 - J_1 - J_6 - J_4 - J_3$, giving zero width. Table 4.6 gives the completion times and the due dates according to the schedule obtained where it can be seen that all the jobs except J_5 are tardy, with respect to both the limits of the completion times. Hence, the number of tardy jobs fall in an interval [5,5].

4.5.1.2.c Objectives 4: Minimizing n_T^l given n_T^u is minimum

Objective 4 is to minimize n_T^l given that n_T^u is minimum. We can obtain minimum number of tardy jobs with respect to the upper limit of the processing times, by applying Moore's Algorithm. Let the number of tardy jobs obtained be n_T^*. We add a constraint restricting $\sum_{i=1}^{n} TU_i \le n_T^*$. The following integer program will give minimum n_T^l keeping n_T^u at its minimum.

$$Minimize \sum_{i=1}^{n} TL_i$$

subject to

Constraints (4.8) to (4.18)

$$\sum_{i=1}^{n} TU_i \le n_T^*$$

$$X_{ij}, TL_i, TU_i \ = \ 0 \ or \ 1$$

This integer program will also require the same number of variables as in the case of Objective 3. An additional constraint is added to restrict the sum of tardy jobs to be the minimum obtained in Objective 4, and hence the number of constraints will be one more than that required to solve for Objective 3.

Example 4.5.1.2

We consider the data given in Table 4.5 for Example 4.5.1.1 and solve for Objective 4 . The integer program for this example will be same as that of Example 4.5.1.1 with a new objective function $Minimize\ \sum_{i=1}^{6} TL_i$ and an additional constraint $\sum_{i=1}^{6} TU_i \leq n_T^*$. The solution for Objective 2 is found to be [3,3], with sequence $J_2 - J_5 - J_4 - J_6 - J_1 - J_3$ and hence $n_T^* = 3$. The solution for Objective 4 obtained from this linear program gives the schedule $J_2 - J_4 - J_3 - J_5 - J_1 - J_6$, giving the solution of [2,3]. Table 4.7 gives the completion times and the due dates according to the schedule obtained where it can be seen that jobs J_1, and J_6 are tardy with respect to lower limits and jobs J_1, J_5 and J_6 are tardy with respect to the upper limits of the completion times.

Table 4.7: Solution for Example 4.5.1.2

Schedule	2	4	3	5	1	6
Completion times	[7,8]	[12,15]	[21,29]	[36,48]	[48,63]	[58,79]
Due dates(d_i)	12	35	35	36	16	36

4.5.1.2.d Objectives 5: Obtaining number of tardy jobs in a desired interval $[n_d^l, n_d^u]$

Objective 5 is to obtain a solution which has the number of tardy jobs in a desired interval $[n_d^l, n_d^u]$. This can be achieved by solving the following system.

$$\text{Constraints (4.8) to (4.18)}$$

$$\sum_{i=1}^{n} TL_i \geq n_d^l$$

$$\sum_{i=1}^{n} TU_i \leq n_d^u$$

$$X_{ij}, TL_i, TU_i \;=\; 0 \text{ or } 1$$

We require the same number of variables as in the case of Objective 3, but the number constraints will increase by 2.

Example 4.5.1.3

Consider the same 6 job problem given in Table 4.5 for Example 4.5.1.1. We are solving for a solution where the number of tardy jobs should fall in an interval [4,5]. The system for this example will be same set of constraints as that of Example 4.5.1.1 with two added constraints $\sum_{i=1}^{6} TL_i \geq 4$ and $\sum_{i=1}^{6} TU_i \leq 5$. The result obtained from this system gives the schedule $J_6 - J_4 - J_2 - J_1 - J_5 - J_3$, giving the solution of [4,4] for the number of tardy jobs. Table 4.8 gives the completion times and the due dates according to the schedule obtained. It can be seen that jobs J_1, J_2, J_3 and J_5 are tardy both with respect to lower and upper limits of completion times.

Table 4.8: Solution for Example 4.5.1.3

Schedule	6	4	2	1	5	3
Completion times	[10,16]	[15,23]	[22,31]	[34,46]	[49,65]	[58,79]
Due dates(d_i)	36	35	12	16	36	35

4.5.1.3 Objective 6: Robust Schedule for minimum number of tardy jobs

The method of obtaining a robust schedule is already explained in the beginning of this chapter. A program in C language was developed which will find out the robust schedule for minimum number of tardy jobs problem. In this code all possible sequences were considered for all the possible combinations of the processing times. It was evident through simulation that the robust schedule can be obtained through extreme values where $p_i = p_i^l$ *or* p_i^u. However, there is no proof to support the observation.

Example 4.5.1.4

Consider a 4 job problem with processing times and respective due dates as given in Table 4.9.

Table 4.9: Data for Example 4.5.1.4

jobs	1	2	3	4
$[p_i^l, p_i^u]$	[3,6]	[8,10]	[5,12]	[6,7]
Due dates(d_i)	15	12	14	20

The robust schedule obtained for this problem is $J_1 - J_4 - J_2 - J_3$, with 2 tardy jobs (J_2 and J_3), and point processing times 6, 8, 12 and 6 for jobs 1, 2, 3 and 4 respectively.

4.5.2 Approach II

In Approach I we considered either lower or upper limits of processing times, to find if a job was tardy. In case of interval processing times the completion time of job i will fall in an interval $[C_i^l, C_i^u]$. If d_i is the due date of job i, it can be ahead of C_i^u, fall some where between C_i^l and the C_i^u, or can be before the C_i^l of a particular job.

We define job J_i as *totally tardy* if $C_i^l > d_i$, *potentially tardy* if $C_i^u > d_i \geq C_i^l$ and *early* if $C_i^u \leq d_i$.

Consider a three job sequence $J_1 - J_2 - J_3$, and the due date as d. It can be seen that job 1 can never be tardy because the due date is greater than the upper limit of the completion time. In the case of job 2 we can see that as per the lower limit of its completion time this job is not tardy and as per the upper limit it is tardy. This satisfies the condition for being a potentially tardy job. In the case of job 3 we

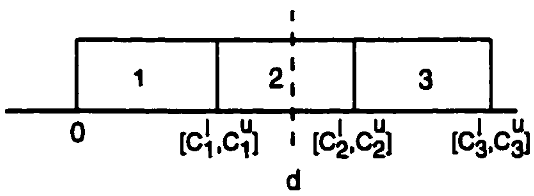

Figure 4.1: Potentially Tardy and Totally Tardy Jobs

can see that the lower limit of its completion time has crossed the due date, which means that this job is a totally tardy job.

Now, that we have defined the terms totally tardy and potentially tardy jobs we would like to discuss different problems that can be formed. We can solve problems like

1. minimizing the number of totally tardy jobs
2. minimizing the number of potentially tardy jobs
3. obtaining totally tardy jobs in a desired interval.
4. minimizing number of potentially tardy jobs given that the totally tardy jobs are minimum and
5. obtaining potentially tardy jobs in a desired interval.

The first two problems can be solved easily using Moore's algorithm. We solve the remaining problems using integer programming.

4.5.2.1 Minimizing the number of Totally Tardy jobs and Potentially Tardy jobs

We know that Moore's algorithm gives the optimal solution for minimizing the number of tardy jobs [49]. We now propose extensions of Moore's algorithm to minimize the number of totally tardy jobs and potentially tardy jobs. The lower limit of the processing time is used to determine if s job is totally tardy. Hence we use the lower limits of the processing times and solve this problem using Moore's algorithm. To minimize the number of potentially tardy jobs we propose a modified Moore's algorithm, given as Algorithm 4.5. This algorithm gives the optimal schedule which minimizes the number of potentially tardy jobs.

algorithm Minimizing Number of Potentially Tardy Jobs;

Step 1: Sequence the jobs according to the EDD rule to find the *current sequence* $J_{(1)}, J_{(2)}, ..., J_{(n)}$ such that
$d_{(k)} \leq d_{(k+1)}$ for $k = 1, 2, .., n-1$.

Step 2: Find the first potentially tardy job say $J_{(k)}$ (when $C_k^l \leq d < C_k^u$). Let this job be $J_{i(l)}$ in the current sequence. If no such job is found, go to Step 4.

Step 3: Find the job in the sequence $J_{(1)}, J_{(2)}, ..., J_{(l)}$, with the largest upper limit of processing time and reject this from the current sequence (select one with maximum lower limit in case of ties). Return to step 2 with a current sequence one shorter than before.

Step 4: Form an optimal schedule by taking the current sequence and appending to it the rejected jobs, which may be sequenced in any order.

NB: This algorithm gives the optimal schedule and not the number of potentially tardy jobs.

Algorithm 4.5: Algorithm for minimizing the number of potentially tardy jobs

Example 4.5.2.1

Consider a 6 job problem with processing times and due dates as given in Table 4.10. The optimal sequence for this problem is $J_4 - J_3 - J_6 - J_2 - J_1 - J_5$, and there are no potentially tardy jobs. The calculations as we apply the algorithm are shown in Table 4.11.

Moore's Algorithm cannot be used to solve the problems 3, 4 and 5 defined

Table 4.10: Data for Example 4.5.2.1

jobs	1	2	3	4	5	6
$[p^l, p^u]$	[12,15]	[7,8]	[9,14]	[5,7]	[15,19]	[1016]
d_i	17	10	25	12	42	40

Table 4.11: Applying Moore's Algorithm

EDD Schedule	2	4	1	3	6	5	Rejected jobs
$duedates(d_i)$	10	12	17	25	40	42	
$[p^l, p^u]$	[7,8]	[5,7]	[12,15]	[9,14]	[10,16]	[15,19]	
Comp.Time	[7,8]	[12,15]					2
Comp.Time	*	[5,7]	[17,22]				1
Comp.Time	*	[5,7]	*	[14,21]	[24,37]	[39,56]	5
Comp.Time	*	[5,7]	*	[14,21]	[24,37]	*	

above. We hence apply the integer programming approach used in Section 4.5.1.2.a.

4.5.2.2 Obtaining the number of totally Tardy Jobs in a desired interval

To solve the problem of obtaining the totally tardy jobs in a desired interval $[n_d^l, n_d^u]$ we solve the system given below. The constraints are given in Section 4.5.1.2.a.

$$\text{Constraints } (4.8), (4.9), (4.10), (4.12) \text{ to } (4.16), \text{ and}$$

$$\sum_{i=1}^{n} TL_i \geq n_d^l$$

$$\sum_{i=1}^{n} TL_i \leq n_d^u$$

$$X_{ij}, TL_i \;=\; 0 \text{ or } 1$$

This system will require $\frac{1}{2}[n(n-1)]^2 + n^2 + 2n$ variables and $\frac{3}{2}[n(n-1)]^2 + 5n + 2$ constraints to solve a problem with n jobs. For a 6 job problem, we need as many as 498 variables and 1382 constraints to solve this problem.

Example 4.5.2.2

Consider the following problem with the data as given in Table 4.12. The problem is to find a schedule which will give the number of totally tardy jobs in the interval [4,5]. The system for this example is given in Appendix II. The α value was taken as 100 and the ϵ value taken was 0.01.

Table 4.12: Data for Example 4.5.2.2

jobs	1	2	3	4	5	6
$[p^l, p^u]$	[12,15]	[7,8]	[9,14]	[5,7]	[15,19]	[1016]
d_i	17	8	15	7	25	32

In the system given in Appendix I, the three constraints (a), (b) and (c) should be added for each of the variable Z_{ijkl} appearing in any constraint. The result obtained from this linear program gives the schedule $J_3 - J_2 - J_6 - J_4 - J_5 - J_1$, providing four totally tardy jobs. Table 4.13 gives the completion times and the due dates according to the schedule obtained where it can be seen that jobs J_1, J_2, J_4 and J_5 are totally tardy.

Table 4.13: Solution for Example 4.5.2.2

Schedule	3	2	6	4	5	1
Completion times	[9,14]	[16,22]	[26,38]	[31,45]	[46,64]	[58,79]
Due dates(d_i)	15	8	32	7	25	17

4.5.2.3 Minimizing number of Potentially Tardy Jobs given Totally tardy jobs are Minimum

To solve the problem of minimizing the number of potentially tardy jobs given that the number of totally tardy jobs is at its minimum, we first have to obtain the minimum number of totally tardy jobs. To get the minimum number of totally tardy jobs we solve the problem using Moore's Algorithm using p^l. Let the minimum number of totally tardy jobs obtained be T^*.

Now we introduce two sets of binary variables, for job i being early or potentially tardy.

$$E_i = \begin{cases} 1, & \text{if } C_i^u \leq d_i \\ 0 & \text{otherwise} \end{cases}$$

$$PT_i = \begin{cases} 1, & \text{if } C_i^l \leq d_i < C_i^u \\ 0 & \text{otherwise} \end{cases}$$

In addition to constraints (4.8) to (4.18) given in Section 4.5.1.2.a we add the following three constraints

$$\alpha E_i \geq d_i - C_i^u + \epsilon \tag{4.19}$$

$$\alpha(1 - E_i) \geq C_i^u - d_i \tag{4.20}$$

$$E_i + TL_i + PT_i = 1 \tag{4.21}$$

Where α is a large number and ϵ is a very small positive number. Constraint (4.19) and (4.20) will make job i early if the upper limit of the completion time of the job is less than the due date. If the job is neither early nor totally tardy constraint (4.21) will make it potentially tardy and assign the binary variable PT_i a value 1. The integer program given below will obtain the minimum number of potentially tardy jobs while the totally tardy jobs are minimum.

$$Minimize \sum_{i=1}^{n} PT_i$$

subject to

Constraints (4.8) to (4.16) and (4.19) to (4.21)

$$\sum_{i=1}^{n} TL_i \leq T^*$$

$$X_{ij}, E_i, TL_i, PT_i = 0 \text{ or } 1$$

Since we are adding new variables PT_i, E_i and C_i^u, the number of variables and constraints to solve the problem will increase. Hence, for an n job problem it will require $\frac{1}{2}[n(n-1)]^2 + n^2 + 5n$ variables and $\frac{3}{2}[n(n-1)]^2 + 9n + 1$ constraints to solve the problem. A 6 job problem will require 516 variables and 1405 constraints.

Example 4.5.2.3

Consider the data given in Table 4.12 for Example 4.5.2.2. The problem is to find a schedule which will have minimum number of potentially tardy jobs given the totally tardy jobs are minimum. The minimum number of totally tardy jobs obtained by applying Moore's algorithm for this data is found to be 3 jobs. The integer program for this example is given in Appendix III.

Table 4.14: Solution for Example 4.5.2.3

Schedule	2	5	6	3	4	1
Completion times	[7,8]	[22,27]	[32,43]	[41,57]	[46,64]	[58,79]
Due dates(d_i)	8	25	32	15	7	17

The schedule obtained by this integer program is $J_2 - J_5 - J_6 - J_3 - J_4 - J_1$. Table 4.14 gives the completion times and the due dates according to the schedule obtained where it can be seen that jobs J_5 and J_6 are potentially tardy while, jobs J_1, J_3 and J_4 are totally tardy.

4.5.2.4 Obtaining the number of Potentially Tardy Jobs in a desired interval

To obtain the number of potentially tardy jobs in a desired interval $[PT_d^l, PT_d^u]$ we solve the following system.

Constraints (4.8) to (4.16) and (4.19) to (4.21)

$$\sum_{i=1}^{n} PT_i \geq PT_d^l$$

$$\sum_{i=1}^{n} PT_i \leq PT_d^u$$

$$X_{ij}, E_i, TL_i, PT_i = 0 \text{ or } 1$$

This system will require the same number of variables but one constraint more than that required for the problem of minimizing the number of potentially tardy jobs given the totally tardy jobs are minimum.

Example 4.5.2.4

Consider the data given in Table 4.12 for Example 4.5.2.2. The problem is to find a schedule which will give number of potentially tardy jobs in a desired interval [1,2]. The constraints of linear program for Example 4.5.2.3 can be used by removing one constraint which restricts the sum of totally tardy jobs to be less than 3 and adding two constraints $\sum_{i=1}^{6} PT_i \geq 1$ and $\sum_{i=1}^{6} PT_i \leq 2$.

The schedule given by this integer program is $J_2 - J_5 - J_4 - J_6 - J_1 - J_3$. Table 4.15 gives the completion times and the due dates according to the schedule obtained where it can be seen that job J_5 is the only potentially tardy.

Table 4.15: Solution for Example 4.5.2.4

Schedule	2	5	4	6	1	3
Completion times	[7,8]	[22,27]	[27,34]	[37,50]	[49,65]	[58,79]
Due dates(d_i)	8	25	7	32	17	15

4.6 Minimizing the sum of Earliness and Tardiness of jobs $\Sigma_{i=1}^{n}(E_i + T_i)$

In this section we consider a single machine scheduling problem when a common due date has to be selected such that the sum of earliness and tardiness is minimized. This problem has been studied by Kanet [25], Sunderaraghavan and Ahmad [52], and Bagchi et al [3].

This problem is now considered for the interval processing times. To obtain a common due date which will minimize the cost we have to consider both the limits of completion times. These values are to be arranged in ascending order. This string has $2n$ values. Hence, we will get two median values and these are the limits of the common due date for the interval processing times case. The proof for the above arguement is given below.

Theorem 4.6.1:

Let $[C_i^l, C_i^u]$ be the interval completion time of the job i. Arrange the values C_i^l, C_i^u for $1 \leq i \leq n$ in increasing order. Let m_i, $i = 1, 2, ..2n$, be the resulting sequence of values.

The minimum of $\sum_{i=1}^{n}(E_i + T_i)$, is achieved if the common due date is selected in the interval $[m_n, m_{n+1}]$.

Proof:

First, we classify the jobs into three sets.

Let $O_i = \{j \mid C_i^l \leq m_i, C_i^u \geq m_{i+1}\}$

$E_i = \{j \mid C_i^u < m_i\}$

$T_i = \{j \mid C_i^l > m_{i+1}\}$

A job in O_i will start before m_i and will finish no earlier than m_{n+1}, while a job in E will finish before m_i, so it is never tardy. Finally, a job in T_i will never be early.

Let K_{O_i}, K_{E_i} and K_{T_i} be the number of jobs in the sets O_i, E_i and T_i respectively. Consider a due date $d \in [m_i, m_{i+1}]$, choose a positive scalar such that $(d + \Delta) \in [m_i, m_{i+1})$.

Let $Z_j(d) = E_j(d) + T_j(d) = [Z_j^l(d), Z_j^u(d)]$ be the sum of earliness and tardiness of job j if the due date is d. Then,

$$Z_j(d) = \begin{cases} [0,(d - C_j^l)] + [0,(C_j^u - d)] & j \in O_i \\ [d - C_j^u, d - C_j^l] & j \in E_i \\ [C_j^l - d, C_j^u - d] & j \in T_i \end{cases}$$

Next we compute $Z_j(d+\Delta)$ and find the relation between $Z_j(d)$ and $Z_j(d+\Delta)$.

$$Z_j(d+\Delta) = \begin{cases} [0,(C_j^u - C_j^l)] = Z_j(d) & j \in O_i \\ [d + \Delta - C_j^u, d + \Delta - C_j^l] = Z_j(d) + \Delta & j \in E_i \\ [C_j^l - d - \Delta, C_j^u - d - \Delta] = Z_j(d) - \Delta & j \in T_i \end{cases}$$

Next, consider the total cost of all jobs.

$$\begin{aligned} Z(d+\Delta) &= \textstyle\sum_{j=1}^{n} Z_j(d+\Delta) \\ &= \textstyle\sum_{j\in O_i} Z_j(d+\Delta) + \sum_{j\in E_i} Z_j(d+\Delta) + \sum_{j\in T_i} Z_j(d+\Delta) \\ &= \textstyle\sum_{j\in O_i} Z_j(d) + \sum_{j\in E_i} Z_j(d) + K_{E_i}\Delta + \sum_{j\in T_i} Z_j(d) - K_{T_i}\Delta \\ &= Z(d) + \Delta(K_{E_i} - K_{T_i}) \end{aligned}$$

If $i = n$, then $K_{E_n} = K_{T_n}$ and from (1) $Z(d+\Delta)$ =Z(d)

If $i > n$, then $K_{E_i} > K_{T_i}$ and $Z(d+\Delta)$ >Z(d)

If $i < n$, then $K_{E_i} < K_{T_i}$ and $Z(d+\Delta)$ <Z(d)

Hence the total cost function is decreasing (negative slope) in the interval $[-\infty, m_n]$, constant (zero slope) in the interval $[m_n, m_{n+1}]$ and increasing (positive slope) in the interval $[m_n, -\infty]$. Hence $d \in [m_n, m_{n+1}]$ gives the solution.

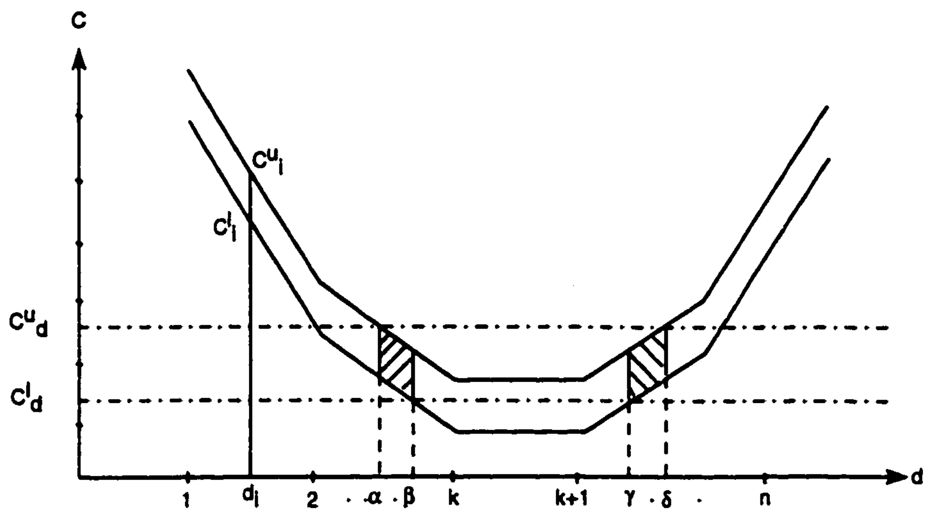

Figure 4.2: Cost Vs Due date Plot

The cost vs due date plot is given in Figure 4.2. If the cost lies in the interval $C = [C^l, C^u]$ the upper curve in the plot represents C^u and the lower curve represents C^l. The solution obtained gives the minimum lower and upper cost limits and hence the first four objectives of the decision maker are achieved simultaneously. The Objective 5 for the interval $[C^l, C^u]$ can be achieved at two due date intervals viz.,

$[\alpha, \beta]$ and $[\gamma, \delta]$. This is clear from the Figure 4.2.

Example 4.6.1

Find a common due date to minimize the sum $\sum_{i=1}^{n}(E_i + T_i)$, for a four job problem whose given sequence is $J_1 - J_2 - J_3 - J_4$. The processing times and the completion times of each job is given in Table 4.16. Now arranging all C_i^l's and C_i^u's in ascending

Table 4.16: Data for Example 4.6.1

jobs	1	2	3	4
$[p_i^l, p_i^u]$	[5,9]	[2,3]	[1,2]	[7,11]
$[C_i^l, C_i^u]$	[5,9]	[7,12]	[8,14]	[15,25]

order we get, $m = \{5, 7, 8, 9, 12, 14, 15, 25\}$

The function C^l, C^u is horizontal for $m_4 \leq d \leq m_5$ i.e. $9 \leq 12$. The sum $\sum_{i=1}^{4}(E_i + T_i)$, for all the due dates in this interval is $[6, 31]$.

4.7 Minimizing the sum $\sum_{i=1}^{n} \alpha_i(E_i + T_i)$

In this section we discuss minimizing the function $I = \sum_{i=1}^{n} \alpha_i(E_i + T_i)$ for a given sequence by finding a common due date. In a given time a job can be either early, potentially tardy or totally tardy depending on its completion time. Let us consider these three situations and calculate the sum of the earliness and tardiness for a job k. Let the completion time of this job lie in the interval $[C_k^l, C_k^u]$, the due date be d and w_k be the width of the completion time interval, $w_k = C_k^u - C_k^l$.

Figure 4.22 shows the situation where job k is early and hence the earliness of this job is $E_k = [d - C_k^u, d - C_k^l]$.

This can be rewritten as

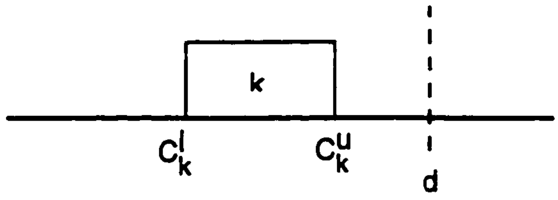

Figure 4.3: Early Job Case

$E_k = [d - C_k^u, d - (C_k^u - w_k)] = [d - C_k^u, d - C_k^u] + [0, w_k]$

The tardiness of this job is $[0, 0]$. Hence

$$E_k + T_k = [d - C_k^u, d - C_k^u] + [0, w_k] \tag{4.22}$$

Figure 4.23 shows the situation where job k is totally tardy and hence the tardiness of this job can be calculated as $T_k = [C_k^l - d, C_k^u - d]$.

This equation can be rewritten as

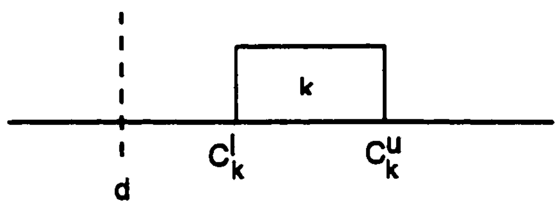

Figure 4.4: Totally Tardy Job Case

$T_k = [C_k^l - d, (C_k^l + w_k) - d] = [C_k^l - d, C_k^l - d] + [0, w_k]$

The earliness of this job is $[0,0]$. Hence

$$E_k + T_k = [C_k^l - d, C_k^l - d] + [0, w_k] \tag{4.23}$$

Finally Figure 4.24 shows the situation where job k is potentially tardy and hence the job can be either early or tardy. The earliness and tardiness of this job are $E_k = [0, d - C_k^l]$ and $T_k = [0, C_k^u - d]$ respectively.

Hence

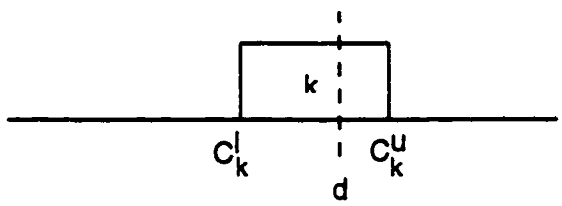

Figure 4.5: Potentially Tardy Job Case

$$E_k + T_k = [0, d - C_k^l] + [0, C_k^u - d] = [0, d - C_k^l + C_k^u - d] = [0, w_k] \tag{4.24}$$

Equations (4.22), (4.23) and (4.24) can be put into a general form for any job k in any of the three situations to find the sum of earliness and tardiness as follows

$$E_k + T_k = max\{d - C_k^u, C_k^l - d, 0\}[1,1] + [0, w_k] \tag{4.25}$$

Consider a schedule S with n jobs and let the common due date be d. Let the weights of these jobs be $\alpha_1, \alpha_2, .., \alpha_n$ respectively. Hence

$$\sum_{i=1}^{n} \alpha_i(E_i + T_i) = \sum_{i=1}^{n} \alpha_i[max\{d - C_i^u, C_i^l - d, 0\}[1,1] + [0, w_i]] \tag{4.26}$$

Given a due date d, let $S_E(d)$, $S_P(d)$ and $S_T(d)$ be the set of early, potentially tardy and totally tardy jobs, then

$$I(d) = \sum_{i=1}^{n} \alpha_i(E_i + T_i)$$

$$= \sum_{i \in S_E(d)} \alpha_i(d - C_i^u)[1,1] + \sum_{i \in S_T(d)} \alpha_i(C_i^l - d)[1,1] + \sum_{i=1}^{n} \alpha_i[0, w_i]$$

$$= d(\sum_{i \in S_E(d)} \alpha_i - \sum_{i \in S_T(d)} \alpha_i)[1,1] + [\sum_{i \in S_T(d)} \alpha_i C_i^l - \sum_{i \in S_E(d)} \alpha_i C_i^u][1,1]$$

$$+ [0, \sum_{i=1}^{n} \alpha_i w_i] \tag{4.27}$$

Let

$$A(d) = d(\sum_{i \in S_E(d)} \alpha_i - \sum_{i \in S_T(d)} \alpha_i)$$

$$B(d) = \sum_{i \in S_T(d)} \alpha_i C_i^l - \sum_{i \in S_E(d)} \alpha_i C_i^u$$

Hence,

$$I(d) = [A(d) + B(d), A(d) + B(d) + \sum_{i=1}^{n} \alpha_i w_i] \tag{4.28}$$

$$= [I^l, I^u]$$

Next we examine the shape of $I(d)$. Note that if $d < min_{1 \le i \le n} C_i^l$, then all the jobs are tardy hence $S_{(E)} = \phi$ and $A(d) < 0$. On the other hand if $d > max_{1 \le i \le n} C_i^u$, then all the jobs are early hence $S_{(T)} = \phi$ and $A(d) > 0$.

Consider d_1, d_2 where $d_2 > d_1$. Then $S_{(E)}(d_1) \subseteq S_{(E)}(d_2)$ and $S_{(T)}(d_2) \subseteq S_{(T)}(d_1)$.

Hence

$$\sum_{i \in S_{(E)}(d_1)} \alpha_i \le \sum_{i \in S_{(E)}(d_2)} \alpha_i$$

$$\sum_{i \in S_{(T)}(d_2)} \alpha_i \leq \sum_{i \in S_{(T)}(d_1)} \alpha_i$$

And one obtains

$$A(d_1) = d_1(\sum_{i \in S_{(E)}(d_1)} \alpha_i - \sum_{i \in S_{(T)}(d_1)} \alpha_i) \leq d_2(\sum_{i \in S_{(E)}(d_2)} \alpha_i - \sum_{i \in S_{(T)}(d_2)} \alpha_i) = A(d_2)$$

Hence we have shown that the slope of I is non-decreasing. In fact it changes from negative to positive. It can be shown that I is a piecewise linear function. Its minimum is achieved at d^* where $A(d^* - \epsilon) < 0$ and $A(d^* + \epsilon) > 0$ for sufficiently small ϵ.

The general method of finding a common due date for minimizing the sum of weighted earliness and tardiness problem can be summarized in four steps.

1. Compute the completion times $[C_i^l, C_i^u]$ for each job i for a given schedule S.

2. Place all the values C_i^l's and C_i^u's considered together in increasing order.

3. Compute the slope of the function by taking the difference of the sum of weights of the early jobs and the sum of weights of the totally tardy jobs for each of the intervals in the critical region.

4. The point d^*, where there is a change in the sign of slope, is the common due date for the schedule S.

The solution obtained is also solution for four objectives of the decision maker. The lower limit of $I(d^*)$ is the optimal solution for Objective 1 and the upper limit of $I(d^*)$ is the optimal for Objective 2. Note that the width of I, $W_I = \sum_{1\leq i\leq n} \alpha_i w_i$, hence any d solves Objective 3. As Objective 2 has an unique solution, the same is the solution for Objective 4. Objective 5 is to achieve a solution in a given interval. This can be achieved from the graph after plotting cost against the due date. The cost vs due date plot is given in Figure 4.6. If the cost lies in the interval $C = [C^l, C^u]$ the upper curve in the plot represents C^u and the lower curve represents C^l. The Objective 5 for the desired interval $[C_d^l, C_d^u]$ can be achieved at two due date intervals viz., $[\alpha, \beta]$ and $[\gamma, \delta]$ as shown in the Figure 4.6.

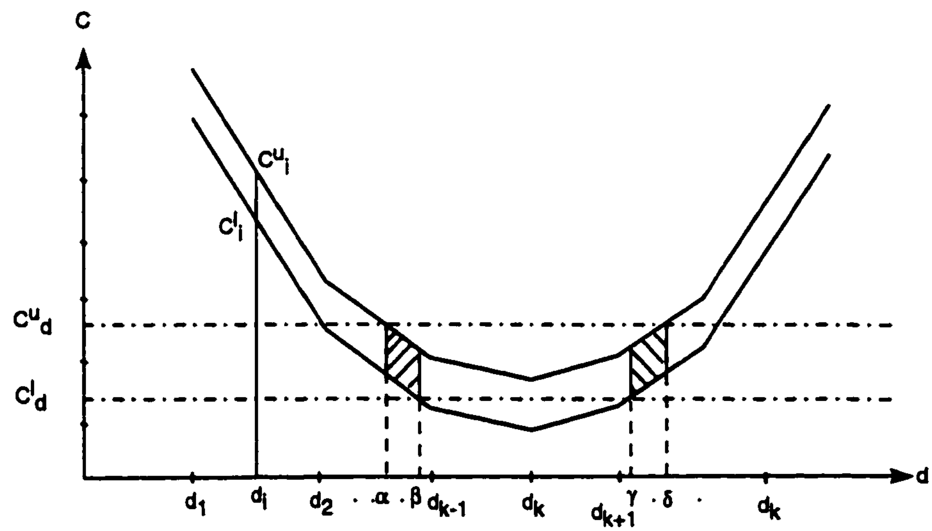

Figure 4.6: Graph for obtaining Objective 5

Example 4.7.1

Finding common due date to minimize the sum $\sum_{i=1}^{n} \alpha_i(E_i+T_i)$ for a five job problem whose given sequence is $J_1 - J_2 - J_3 - J_4 - J_5$. The processing times, completion times and the weights of each job are given in Table 4.17.

Table 4.17: Data for Example 4.7.1

jobs	1	2	3	4	5
$[p_i^l, p_i^u]$	[5,15]	[7,20]	[6,18]	[10,13]	[8,17]
$[C_i^l, C_i^u]$	[5,15]	[12,35]	[18,53]	[28,66]	[36,83]
w_i	4	3	1	2	5

Arranging all C_i^l's and C_i^u's in ascending order we get,

$$\{5, 12, 15, 18, 28, 35, 36, 53, 66, 83\}$$

Now, finding the difference $\sum_{i=S_E(d)}^{n} \alpha_i - \sum_{i=S_T(d)}^{n} \alpha_i$ for each of the interval

$$\{0-5, 5-12, 12-15, 15-18, 18-28, 28-35, 35-36, 36-53, 53-66, 66-83, 83\, \textit{and above}\}$$

we get the string

$$\{-15, -11, -8, -4, -3, -1, 2, 7, 8, 10, 15\}$$

The point where these values change from negative to positive is the solution for this problem. The due date is the corresponding completion time value at which the slope changes and the value is 35. The common due date at which weighted sum of earliness and tardiness is minimum, for the jobs with different weights for this example is 35. The calculated value for the sum $\sum_{i=1}^{n} \alpha_i(E_i + T_i)$ is $[85, 540]$.

Chapter 5

Summary & Conclusions

5.1 Summary

In this research we have defined a new class of problems where the parameters of the optimization problems are given as intervals. We have solved various network and scheduling problems, where the parameters are given in the form of intervals. Table 5.1 gives the list of both the network and scheduling problems we have solved. Table 5.1 also gives the details of methods applied to achieve various objectives of the decision maker (mentioned in Section 2.4).

In Chapter 3 we solved the project management problem using critical path method (CPM), crashing of network problem, minimal spanning tree problem, transportation problem and assignment problem. We solved these network problems for various objectives of the decision maker. In the case of CPM we gave algorithms

Table 5.1: Summary of the problems solved for different objectives

Problems	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Obj. 5	Obj. 6
Critical Path Method	Algm	Algm	Algm	Algm	N/A	Bintong[2]
Crashing Networks	LP	LP	LP	LP	LP	
Minimal Spanning Tree	Algm	Algm	Algm	Algm	IP	
Transportation	LP	LP	LP	Algm	LP	
Assignment	Algm	Algm	Algm	Algm	IP	
Mean Flow time	Algm	Algm	Algm	Algm	IP	Algm
Min. the Max. Lateness	Algm	Algm	Algm	Algm	Algm	Algm
Min. No. of Tardy jobs	Algm	Algm	IP	IP	IP	Heuristic
Min $\sum E_i + \sum T_i$	Algm	Algm	Algm	Algm	Algm	
Min $\sum_{i=1}^{n} \alpha_i(E_i + T_i)$	Algm	Algm	Algm	Algm	Algm	

Algm = Algorithm,LP = Linear Program, IP = Integer Program.

to achieve first four objectives and we showed that Objective 5 is not applicable in this case. The crashing of network was formulated as a linear program. For the minimal spanning tree problem we gave algorithms for the first four objectives and integer program formulation for the fifth objective. Transportation problem was solved using linear programming for Objectives 1,2,3 and 5. An algorithmic procedure was given to solve the transportation problem for Objective 4. For the assignment problem algorithms were provided to solve for the first four objectives and an integer program was developed for the fifth objective.

In Chapter 4 we solved various scheduling problems, viz., minimizing mean flow time problem, minimizing the maximum lateness problem, minimizing the number of tardy jobs problem and minimizing the sum of earliness and tardiness for all jobs, with and without weights for a given schedule. In this chapter, we introduced the sixth objective of the decision maker i.e., to find a robust schedule for any performance measure. For the mean flow time problem we have given algorithms to solve first four objectives of the decision maker and a linear program for the fifth objective. A C program is developed to find the robust schedule for mean flow time problem. For the maximum lateness problem we have proved that the earliest due date (EDD) scheduling provides with the unique solution for all the objectives including the robust schedule.

Two approaches are proposed to solve the minimum number of tardy jobs problem. Approach I is to consider either lower or the upper limits of the completion times to find if a job is tardy or not. In Approach I we obtain the number of tardy jobs in the form of an interval. We extended the Moore's algorithm to solve this problem for the first two objectives. The other three objecitves were solved using integer programming. In Approach II the completion time intervals itself were considered and hence there was a need to redefine the tardiness of a job. We redefined the tardy jobs into two categories, viz., totally tardy jobs and potentially tardy jobs. Algorithms were given for minimizing the totally tardy jobs and potentially tardy jobs. The problem of minimizing the potentially tardy jobs given that the totally tardy jobs are minimized, was formulated as an integer program. Integer formulations were given for obtaining solutions in desired interval for the totally tardy and potentially tardy jobs. A Heuristic was proposed to get a robust schedule for the minimizing the number of tardy jobs problem. A method was proposed to obtain the common due date, which minimizes the sum of earliness and tardiness for all jobs in a given schedule. Finally, a method was proposed to obtain a common due date which minimizes the weighted sum of earliness and tardiness for jobs with different weights, for a given schedule. The processing times were considered as interval values in both the cases.

5.2 Further Research and Extensions

This thesis provides the base for solving the proposed class of problems, where the parameters of the optimization problems are given as intervals. This research has provided algorithms and formulations as linear or integer programs for different problems. Various network and scheduling problems have been solved. With this research we have opened doors for more extensive research by adding more problems to this class by considering interval valued parameters.

This type of research can be extended to solve more network problems, like, maximum flow problems, minimal flow problems, resource constrained project scheduling problems, etc., In this thesis we have solved only some of the single machine scheduling problems. More research can be done and extended to job shop problems with multiple machines. Efficient algorithms can be developed for all the linear/integer programs which are proposed in this research for different problems.

Appendix I

Integer Program for Example 4.5.1.1

$$Minimize \ \sum_{i=1}^{6} TU_i - \sum_{i=1}^{6} TL_i$$

Subject to

$$C_1^l - 46X_{15} - 46X_{16} - 7Z_{1221} - 9Z_{1231} - 5Z_{1241} - 15Z_{1251} - 10Z_{1261} - 7Z_{1321} - 7Z_{1322}$$
$$-9Z_{1331}-9Z_{1332}-5Z_{1341}-5Z_{1342}-15Z_{1351}-15Z_{1352}-10Z_{1361}-10Z_{1362}-7Z_{1421}-7Z_{1422}$$
$$-7Z_{1423}-9Z_{1431}-9Z_{1432}-9Z_{1433}-5Z_{1441}-5Z_{1442}-5Z_{1443}-15Z_{1451}-15Z_{1452}-15Z_{1453}$$
$$-10Z_{1461}-10Z_{1462}-10Z_{1463}+7Z_{1525}+7Z_{1526}+9Z_{1535}+9Z_{1536}+5Z_{1545}+5Z_{1546}+15Z_{1555}$$
$$+15Z_{1556} + 10Z_{1565} + 10Z_{1566} + 7Z_{1626} + 9Z_{1636} + 5Z_{1646} + 15Z_{1656} + 10Z_{1666} = 12$$

$$C_1^u - 64X_{15} - 64X_{16} - 8Z_{1221} - 14Z_{1231} - 7Z_{1241} - 19Z_{1251} - 16Z_{1261} - 8Z_{1321} - 8Z_{1322}$$
$$-14Z_{1331}-14Z_{1332}-7Z_{1341}-7Z_{1342}-19Z_{1351}-19Z_{1352}-16Z_{1361}-16Z_{1362}-8Z_{1421}-8Z_{1422}$$
$$-8Z_{1423}-14Z_{1431}-14Z_{1432}-14Z_{1433}-7Z_{1441}-7Z_{1442}-7Z_{1443}-19Z_{1451}-19Z_{1452}-19Z_{1453}$$
$$-16Z_{1461}-16Z_{1462}-16Z_{1463}+8Z_{1525}+8Z_{1526}+14Z_{1535}+14Z_{1536}+7Z_{1545}+7Z_{1546}+19Z_{1555}$$
$$+19Z_{1556} + 16Z_{1565} + 16Z_{1566} + 8Z_{1626} + 14Z_{1636} + 7Z_{1646} + 19Z_{1656} + 16Z_{1666} = 15$$

$$C_2^l - 51X_{25} - 51X_{26} - 12Z_{2211} - 9Z_{2231} - 5Z_{2241} - 15Z_{2251} - 10Z_{2261} - 12Z_{2311} - 12Z_{1312}$$
$$-9Z_{2331}-9Z_{2332}-5Z_{2341}-5Z_{2342}-15Z_{2351}-15Z_{2352}-10Z_{2361}-10Z_{2362}-12Z_{2421}-12Z_{2422}$$
$$-12Z_{2423}-9Z_{2431}-9Z_{2432}-9Z_{2433}-5Z_{2441}-5Z_{2442}-5Z_{2443}-15Z_{2451}-15Z_{2452}-15Z_{2453}$$
$$-10Z_{2461}-10Z_{2462}-10Z_{2463}+12Z_{2525}+12Z_{2526}+9Z_{2535}+9Z_{2536}+5Z_{2545}+5Z_{2546}+15Z_{2555}$$
$$+15Z_{2556} + 10Z_{2565} + 10Z_{1566} + 12Z_{2626} + 12Z_{2636} + 5Z_{2646} + 15Z_{2656} + 10Z_{2666} = 7$$

$$C_2^u - 71X_{25} - 71X_{26} - 15Z_{2211} - 14Z_{2231} - 7Z_{2241} - 19Z_{2251} - 16Z_{2261} - 15Z_{2311} - 15Z_{1312}$$
$$-14Z_{2331}-14Z_{2332}-7Z_{2341}-7Z_{2342}-19Z_{2351}-19Z_{2352}-16Z_{2361}-16Z_{2362}-15Z_{2421}-15Z_{2422}$$
$$-15Z_{2423}-14Z_{2431}-14Z_{2432}-14Z_{2433}-7Z_{2441}-7Z_{2442}-7Z_{2443}-19Z_{2451}-19Z_{2452}-19Z_{2453}$$

$$-16Z_{2461}-16Z_{2462}-16Z_{2463}+15Z_{2525}+15Z_{2526}+14Z_{2535}+14Z_{2536}+7Z_{2545}+7Z_{2546}+19Z_{2555}$$
$$+19Z_{2556}+16Z_{2565}+16Z_{1566}+15Z_{2626}+15Z_{2636}+7Z_{2646}+19Z_{2656}+16Z_{2666}=8$$

$$C_3^l-49X_{35}-49X_{36}-12Z_{3211}-7Z_{3231}-5Z_{3241}-15Z_{3251}-10Z_{3261}-12Z_{3311}-12Z_{3312}$$
$$-7Z_{3321}-7Z_{3322}-5Z_{3341}-5Z_{3342}-15Z_{3351}-15Z_{3352}-10Z_{3361}-10Z_{3362}-12Z_{3411}-12Z_{3412}$$
$$-12Z_{3413}-7Z_{3421}-7Z_{3422}-7Z_{3423}-5Z_{3441}-5Z_{3442}-5Z_{3443}-15Z_{3451}-15Z_{3452}-15Z_{3453}$$
$$-10Z_{3461}-10Z_{3462}-10Z_{3463}+12Z_{3515}+12Z_{3516}+7Z_{3525}+7Z_{3526}+5Z_{3545}+5Z_{3546}+15Z_{3555}$$
$$+15Z_{3556}+10Z_{3565}+10Z_{3566}+12Z_{3616}+7Z_{3626}+5Z_{3646}+15Z_{3656}+10Z_{3666}=9$$

$$C_3^u-65X_{35}-65X_{36}-15Z_{3211}-8Z_{3231}-7Z_{3241}-19Z_{3251}-16Z_{3261}-15Z_{3311}-15Z_{3312}$$
$$-8Z_{3321}-8Z_{3322}-7Z_{3341}-7Z_{3342}-19Z_{3351}-19Z_{3352}-16Z_{3361}-16Z_{3362}-15Z_{3411}-15Z_{3412}$$
$$-15Z_{3413}-8Z_{3421}-8Z_{3422}-8Z_{3423}-7Z_{3441}-7Z_{3442}-7Z_{3443}-19Z_{3451}-19Z_{3452}-19Z_{3453}$$
$$-16Z_{3461}-16Z_{3462}-16Z_{3463}+15Z_{3515}+15Z_{3516}+8Z_{3525}+8Z_{3526}+7Z_{3545}+7Z_{3546}+19Z_{3555}$$
$$+19Z_{3556}+16Z_{3565}+16Z_{3566}+15Z_{3616}+8Z_{3626}+7Z_{3646}+19Z_{3656}+16Z_{3666}=14$$

$$C_4^l-53X_{45}-53X_{46}-12Z_{4211}-7Z_{4221}-9Z_{4231}-15Z_{4251}-10Z_{4261}-12Z_{4311}-12Z_{4312}$$
$$-7Z_{4321}-7Z_{4322}-9Z_{4331}-9Z_{4332}-15Z_{4351}-15Z_{4352}-10Z_{4361}-10Z_{4362}-12Z_{4411}-12Z_{4412}$$
$$-12Z_{4413}-7Z_{4421}-7Z_{4422}-7Z_{4423}-9Z_{4431}-9Z_{4432}-9Z_{4433}-15Z_{4451}-15Z_{4452}-15Z_{4453}$$
$$-10Z_{4461}-10Z_{4462}-10Z_{4463}+12Z_{4515}+12Z_{4516}+7Z_{4525}+7Z_{4526}+9Z_{4535}+9Z_{4536}+15Z_{4555}$$
$$+15Z_{4556}+10Z_{4565}+10Z_{4566}+12Z_{4616}+7Z_{4626}+9Z_{4636}+15Z_{4656}+10Z_{4666}=5$$

$$C_4^u-72X_{45}-72X_{46}-15Z_{4211}-8Z_{4221}-14Z_{4231}-19Z_{4251}-16Z_{4261}-15Z_{4311}-15Z_{4312}$$
$$-8Z_{4321}-8Z_{4322}-14Z_{4331}-14Z_{4332}-19Z_{4351}-19Z_{4352}-16Z_{4361}-16Z_{4362}-15Z_{4411}-15Z_{4412}$$
$$-15Z_{4413}-8Z_{4421}-8Z_{4422}-8Z_{4423}-14Z_{4431}-14Z_{4432}-14Z_{4433}-19Z_{4451}-19Z_{4452}-19Z_{4453}$$
$$-16Z_{4461}-16Z_{4462}-16Z_{4463}+15Z_{4515}+15Z_{4516}+8Z_{4525}+8Z_{4526}+14Z_{4535}+14Z_{4536}+19Z_{4555}$$
$$+19Z_{4556}+16Z_{4565}+16Z_{4566}+15Z_{4616}+8Z_{4626}+14Z_{4636}+19Z_{4656}+16Z_{4666}=7$$

$$C_5^l - 43X_{55} - 43X_{56} - 12Z_{5211} - 7Z_{5221} - 9Z_{5231} - 5Z_{5241} - 10Z_{5261} - 12Z_{5311} - 12Z_{5312}$$
$$-7Z_{5321} - 7Z_{5322} - 9Z_{5331} - 9Z_{5332} - 5Z_{5341} - 5Z_{5342} - 10Z_{5361} - 10Z_{5362} - 12Z_{5411} - 12Z_{5412}$$
$$-12Z_{5413} - 7Z_{5421} - 7Z_{5422} - 7Z_{5423} - 9Z_{5431} - 9Z_{5432} - 9Z_{5433} - 5Z_{5441} - 5Z_{5442} - 5Z_{5443}$$
$$-10Z_{5461} - 10Z_{5462} - 10Z_{5463} + 12Z_{5515} + 12Z_{5516} + 7Z_{5525} + 7Z_{5526} + 9Z_{5535} + 9Z_{5536} + 5Z_{5545}$$
$$+5Z_{5546} + 10Z_{5565} + 10Z_{5566} + 12Z_{5616} + 7Z_{5626} + 9Z_{5636} + 5Z_{5646} + 10Z_{5666} = 15$$

$$C_5^u - 60X_{55} - 60X_{56} - 15Z_{5211} - 8Z_{5221} - 14Z_{5231} - 7Z_{5241} - 16Z_{5261} - 15Z_{5311} - 15Z_{5312}$$
$$-8Z_{5321} - 8Z_{5322} - 14Z_{5331} - 14Z_{5332} - 7Z_{5341} - 7Z_{5342} - 16Z_{5361} - 16Z_{5362} - 15Z_{5411} - 15Z_{5412}$$
$$-15Z_{5413} - 8Z_{5421} - 8Z_{5422} - 8Z_{5423} - 14Z_{5431} - 14Z_{5432} - 14Z_{5433} - 7Z_{5441} - 7Z_{5442} - 7Z_{5443}$$
$$-16Z_{5461} - 16Z_{5462} - 16Z_{5463} + 15Z_{5515} + 15Z_{5516} + 8Z_{5525} + 8Z_{5526} + 14Z_{5535} + 14Z_{5536} + 7Z_{5545}$$
$$+7Z_{5546} + 16Z_{5565} + 16Z_{5566} + 15Z_{5616} + 8Z_{5626} + 14Z_{5636} + 7Z_{5646} + 16Z_{5666} = 19$$

$$C_6^l - 48X_{65} - 48X_{66} - 12Z_{6211} - 7Z_{6221} - 9Z_{6231} - 5Z_{6241} - 15Z_{6251} - 12Z_{6311} - 12Z_{6312}$$
$$-7Z_{6321} - 7Z_{6322} - 9Z_{6331} - 9Z_{6332} - 5Z_{6341} - 5Z_{6342} - 15Z_{6351} - 15Z_{6352} - 12Z_{6411} - 12Z_{6412}$$
$$-12Z_{6413} - 7Z_{6421} - 7Z_{6422} - 7Z_{6423} - 9Z_{6431} - 9Z_{6432} - 9Z_{6433} - 5Z_{6441} - 5Z_{6442} - 5Z_{6443}$$
$$-15Z_{6451} - 15Z_{6452} - 15Z_{6453} + 12Z_{6515} + 12Z_{6516} + 7Z_{6525} + 7Z_{6526} + 9Z_{6535} + 9Z_{6536} + 5Z_{6545}$$
$$+5Z_{6546} + 15Z_{6555} + 15Z_{6555} + 12Z_{6616} + 7Z_{6626} + 9Z_{6636} + 5Z_{6646} + 15Z_{6656} = 10$$

$$C_6^u - 63X_{65} - 63X_{66} - 15Z_{6211} - 8Z_{6221} - 14Z_{6231} - 14Z_{6241} - 19Z_{6251} - 15Z_{6311} - 15Z_{6312}$$
$$-8Z_{6321} - 8Z_{6322} - 14Z_{6331} - 14Z_{6332} - 7Z_{6341} - 7Z_{6342} - 19Z_{6351} - 19Z_{6352} - 15Z_{6411} - 15Z_{6412}$$
$$-15Z_{6413} - 8Z_{6421} - 8Z_{6422} - 8Z_{6423} - 14Z_{6431} - 14Z_{6432} - 14Z_{6433} - 7Z_{6441} - 7Z_{6442} - 7Z_{6443}$$
$$-19Z_{6451} - 19Z_{6452} - 19Z_{6453} + 15Z_{6515} + 15Z_{6516} + 8Z_{6525} + 8Z_{6526} + 14Z_{6535} + 14Z_{6536} + 7Z_{6545}$$
$$+7Z_{6546} + 19Z_{6555} + 19Z_{6555} + 15Z_{6616} + 8Z_{6626} + 14Z_{6636} + 7Z_{6646} + 19Z_{6656} = 16$$

$$Z_{ijkl} \leq X_{ij} \qquad (a)$$
$$Z_{ijkl} \leq X_{kl} \qquad (b)$$
$$X_{ij} + X_{kl} - 1 \leq Z_{ijkl} \qquad (c)$$

$$C_1^l - 100TL_1 \leq 16$$
$$C_1^u - 100TU_1 \leq 16$$
$$C_{2^l} - 100TL_2 \leq 12$$
$$C_{2^u} - 100TU_2 \leq 12$$
$$C_3^l - 100TL_3 \leq 35$$
$$C_3^u - 100TU_3 \leq 35$$
$$C_{4^l} - 100TL_4 \leq 35$$
$$C_{4^u} - 100TU_4 \leq 35$$
$$C_5^l - 100TL_5 \leq 36$$
$$C_5^u - 100TU_5 \leq 36$$
$$C_6^l - 100TL_6 \leq 36$$
$$C_6^u - 100TU_6 \leq 36$$

$$100TL_1 - C_1^l \leq 83.99$$
$$100TU_1 - C_1^u \leq 83.99$$
$$100TL_2 - C_2^l \leq 87.99$$
$$100TU_2 - C_2^u \leq 87.99$$
$$100TL_3 - C_3^l \leq 64.99$$
$$100TU_3 - C_3^u \leq 64.99$$
$$100TL_4 - C_4^l \leq 64.99$$
$$100TU_4 - C_4^u \leq 64.99$$
$$100TL_5 - C_5^l \leq 63.99$$
$$100TU_5 - C_5^u \leq 63.99$$
$$100TL_6 - C_6^l \leq 63.99$$
$$100TU_6 - C_6^u \leq 63.99$$

$$X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} = 1$$
$$X_{21} + X_{22} + X_{23} + X_{24} + X_{25} + X_{26} = 1$$
$$X_{31} + X_{32} + X_{33} + X_{34} + X_{35} + X_{36} = 1$$

$$X_{41} + X_{42} + X_{43} + X_{44} + X_{45} + X_{46} = 1$$
$$X_{51} + X_{52} + X_{53} + X_{54} + X_{55} + X_{56} = 1$$
$$X_{61} + X_{62} + X_{63} + X_{64} + X_{65} + X_{66} = 1$$
$$X_{11} + X_{21} + X_{31} + X_{41} + X_{51} + X_{61} = 1$$
$$X_{12} + X_{22} + X_{32} + X_{42} + X_{52} + X_{62} = 1$$
$$X_{13} + X_{23} + X_{33} + X_{43} + X_{53} + X_{63} = 1$$
$$X_{14} + X_{24} + X_{34} + X_{44} + X_{54} + X_{64} = 1$$
$$X_{15} + X_{25} + X_{35} + X_{45} + X_{55} + X_{65} = 1$$
$$X_{16} + X_{26} + X_{36} + X_{46} + X_{56} + X_{66} = 1$$
$$X_{ij}, E_i, TL_i, TU_i, PT_i = \ 0 \ or \ 1$$

Appendix II

System for Example 4.5.2.2

$$TL_1 + TL_2 + TL_3 + TL_4 + TL_5 + TL_6 \geq 4$$

$$TL_1 + TL_2 + TL_3 + TL_4 + TL_5 + TL_6 \leq 5$$

$$C_1^l - 46X_{15} - 46X_{16} - 7Z_{1221} - 9Z_{1231} - 5Z_{1241} - 15Z_{1251} - 10Z_{1261} - 7Z_{1321} - 7Z_{1322}$$
$$-9Z_{1331} - 9Z_{1332} - 5Z_{1341} - 5Z_{1342} - 15Z_{1351} - 15Z_{1352} - 10Z_{1361} - 10Z_{1362} - 7Z_{1421} - 7Z_{1422}$$
$$-7Z_{1423} - 9Z_{1431} - 9Z_{1432} - 9Z_{1433} - 5Z_{1441} - 5Z_{1442} - 5Z_{1443} - 15Z_{1451} - 15Z_{1452} - 15Z_{1453}$$
$$-10Z_{1461} - 10Z_{1462} - 10Z_{1463} + 7Z_{1525} + 7Z_{1526} + 9Z_{1535} + 9Z_{1536} + 5Z_{1545} + 5Z_{1546} + 15Z_{1555}$$
$$+15Z_{1556} + 10Z_{1565} + 10Z_{1566} + 7Z_{1626} + 9Z_{1636} + 5Z_{1646} + 15Z_{1656} + 10Z_{1666} = 12$$

$$C_2^l - 51X_{25} - 51X_{26} - 12Z_{2211} - 9Z_{2231} - 5Z_{2241} - 15Z_{2251} - 10Z_{2261} - 12Z_{2311} - 12Z_{1312}$$
$$-9Z_{2331} - 9Z_{2332} - 5Z_{2341} - 5Z_{2342} - 15Z_{2351} - 15Z_{2352} - 10Z_{2361} - 10Z_{2362} - 12Z_{2421} - 12Z_{2422}$$
$$-12Z_{2423} - 9Z_{2431} - 9Z_{2432} - 9Z_{2433} - 5Z_{2441} - 5Z_{2442} - 5Z_{2443} - 15Z_{2451} - 15Z_{2452} - 15Z_{2453}$$
$$-10Z_{2461} - 10Z_{2462} - 10Z_{2463} + 12Z_{2525} + 12Z_{2526} + 9Z_{2535} + 9Z_{2536} + 5Z_{2545} + 5Z_{2546} + 15Z_{2555}$$
$$+15Z_{2556} + 10Z_{2565} + 10Z_{1566} + 12Z_{2626} + 12Z_{2636} + 5Z_{2646} + 15Z_{2656} + 10Z_{2666} = 7$$

$$C_3^l - 49X_{35} - 49X_{36} - 12Z_{3211} - 7Z_{3231} - 5Z_{3241} - 15Z_{3251} - 10Z_{3261} - 12Z_{3311} - 12Z_{3312}$$
$$-7Z_{3321} - 7Z_{3322} - 5Z_{3341} - 5Z_{3342} - 15Z_{3351} - 15Z_{3352} - 10Z_{3361} - 10Z_{3362} - 12Z_{3411} - 12Z_{3412}$$
$$-12Z_{3413} - 7Z_{3421} - 7Z_{3422} - 7Z_{3423} - 5Z_{3441} - 5Z_{3442} - 5Z_{3443} - 15Z_{3451} - 15Z_{3452} - 15Z_{3453}$$
$$-10Z_{3461} - 10Z_{3462} - 10Z_{3463} + 12Z_{3515} + 12Z_{3516} + 7Z_{3525} + 7Z_{3526} + 5Z_{3545} + 5Z_{3546} + 15Z_{3555}$$
$$+15Z_{3556} + 10Z_{3565} + 10Z_{3566} + 12Z_{3616} + 7Z_{3626} + 5Z_{3646} + 15Z_{3656} + 10Z_{3666} = 9$$

$$C_4^l - 53X_{45} - 53X_{46} - 12Z_{4211} - 7Z_{4221} - 9Z_{4231} - 15Z_{4251} - 10Z_{4261} - 12Z_{4311} - 12Z_{4312}$$
$$-7Z_{4321} - 7Z_{4322} - 9Z_{4331} - 9Z_{4332} - 15Z_{4351} - 15Z_{4352} - 10Z_{4361} - 10Z_{4362} - 12Z_{4411} - 12Z_{4412}$$
$$-12Z_{4413} - 7Z_{4421} - 7Z_{4422} - 7Z_{4423} - 9Z_{4431} - 9Z_{4432} - 9Z_{4433} - 15Z_{4451} - 15Z_{4452} - 15Z_{4453}$$
$$-10Z_{4461} - 10Z_{4462} - 10Z_{4463} + 12Z_{4515} + 12Z_{4516} + 7Z_{4525} + 7Z_{4526} + 9Z_{4535} + 9Z_{4536} + 15Z_{4555}$$

$$+15Z_{4556}+10Z_{4565}+10Z_{4566}+12Z_{4616}+7Z_{4626}+9Z_{4636}+15Z_{4656}+10Z_{4666}=5$$

$$C_5^l-43X_{55}-43X_{56}-12Z_{5211}-7Z_{5221}-9Z_{5231}-5Z_{5241}-10Z_{5261}-12Z_{5311}-12Z_{5312}$$
$$-7Z_{5321}-7Z_{5322}-9Z_{5331}-9Z_{5332}-5Z_{5341}-5Z_{5342}-10Z_{5361}-10Z_{5362}-12Z_{5411}-12Z_{5412}$$
$$-12Z_{5413}-7Z_{5421}-7Z_{5422}-7Z_{5423}-9Z_{5431}-9Z_{5432}-9Z_{5433}-5Z_{5441}-5Z_{5442}-5Z_{5443}$$
$$-10Z_{5461}-10Z_{5462}-10Z_{5463}+12Z_{5515}+12Z_{5516}+7Z_{5525}+7Z_{5526}+9Z_{5535}+9Z_{5536}+5Z_{5545}$$
$$+5Z_{5546}+10Z_{5565}+10Z_{5566}+12Z_{5616}+7Z_{5626}+9Z_{5636}+5Z_{5646}+10Z_{5666}=15$$

$$C_6^l-48X_{65}-48X_{66}-12Z_{6211}-7Z_{6221}-9Z_{6231}-5Z_{6241}-15Z_{6251}-12Z_{6311}-12Z_{6312}$$
$$-7Z_{6321}-7Z_{6322}-9Z_{6331}-9Z_{6332}-5Z_{6341}-5Z_{6342}-15Z_{6351}-15Z_{6352}-12Z_{6411}-12Z_{6412}$$
$$-12Z_{6413}-7Z_{6421}-7Z_{6422}-7Z_{6423}-9Z_{6431}-9Z_{6432}-9Z_{6433}-5Z_{6441}-5Z_{6442}-5Z_{6443}$$
$$-15Z_{6451}-15Z_{6452}-15Z_{6453}+12Z_{6515}+12Z_{6516}+7Z_{6525}+7Z_{6526}+9Z_{6535}+9Z_{6536}+5Z_{6545}$$
$$+5Z_{6546}+15Z_{6555}+15Z_{6555}+12Z_{6616}+7Z_{6626}+9Z_{6636}+5Z_{6646}+15Z_{6656}=10$$

$$Z_{ijkl}\le X_{ij} \qquad (a)$$
$$Z_{ijkl}\le X_{kl} \qquad (b)$$
$$X_{ij}+X_{kl}-1\le Z_{ijkl} \qquad (c)$$

$$C_1^l-100TL_1\le 17$$
$$C_2^l-100TL_2\le 8$$
$$C_3^l-100TL_3\le 15$$
$$C_4^l-100TL_4\le 7$$
$$C_5^l-100TL_5\le 25$$
$$C_6^l-100TL_6\le 32$$

$$100TL_1-C_1^l\le 82.99$$
$$100TL_2-C_2^l\le 91.99$$
$$100TL_3-C_3^l\le 84.99$$
$$100TL_4-C_4^l\le 92.99$$

$$100TL_5 - C_5^l \leq 74.99$$
$$100TL_6 - C_6^l \leq 67.99$$

$$X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} = 1$$
$$X_{21} + X_{22} + X_{23} + X_{24} + X_{25} + X_{26} = 1$$
$$X_{31} + X_{32} + X_{33} + X_{34} + X_{35} + X_{36} = 1$$
$$X_{41} + X_{42} + X_{43} + X_{44} + X_{45} + X_{46} = 1$$
$$X_{51} + X_{52} + X_{53} + X_{54} + X_{55} + X_{56} = 1$$
$$X_{61} + X_{62} + X_{63} + X_{64} + X_{65} + X_{66} = 1$$
$$X_{11} + X_{21} + X_{31} + X_{41} + X_{51} + X_{61} = 1$$
$$X_{12} + X_{22} + X_{32} + X_{42} + X_{52} + X_{62} = 1$$
$$X_{13} + X_{23} + X_{33} + X_{43} + X_{53} + X_{63} = 1$$
$$X_{14} + X_{24} + X_{34} + X_{44} + X_{54} + X_{64} = 1$$
$$X_{15} + X_{25} + X_{35} + X_{45} + X_{55} + X_{65} = 1$$
$$X_{16} + X_{26} + X_{36} + X_{46} + X_{56} + X_{66} = 1$$
$$X_{ij}, TL_i = \ 0 \ or \ 1$$

Appendix III

Linear Program for Example 4.5.2.3

$$Minimize \;\; PT_1 + PT_2 + PT_3 + PT_4 + PT_5 + PT_6$$

Subject to

$$TL_1 + TL_2 + TL_3 + TL_4 + TL_5 + TL_6 \leq 3$$

$$\begin{aligned}
&C_1^l - 46X_{15} - 46X_{16} - 7Z_{1221} - 9Z_{1231} - 5Z_{1241} - 15Z_{1251} - 10Z_{1261} - 7Z_{1321} - 7Z_{1322} \\
&-9Z_{1331} - 9Z_{1332} - 5Z_{1341} - 5Z_{1342} - 15Z_{1351} - 15Z_{1352} - 10Z_{1361} - 10Z_{1362} - 7Z_{1421} - 7Z_{1422} \\
&-7Z_{1423} - 9Z_{1431} - 9Z_{1432} - 9Z_{1433} - 5Z_{1441} - 5Z_{1442} - 5Z_{1443} - 15Z_{1451} - 15Z_{1452} - 15Z_{1453} \\
&-10Z_{1461} - 10Z_{1462} - 10Z_{1463} + 7Z_{1525} + 7Z_{1526} + 9Z_{1535} + 9Z_{1536} + 5Z_{1545} + 5Z_{1546} + 15Z_{1555} \\
&+15Z_{1556} + 10Z_{1565} + 10Z_{1566} + 7Z_{1626} + 9Z_{1636} + 5Z_{1646} + 15Z_{1656} + 10Z_{1666} = 12
\end{aligned}$$

$$\begin{aligned}
&C_1^u - 64X_{15} - 64X_{16} - 8Z_{1221} - 14Z_{1231} - 7Z_{1241} - 19Z_{1251} - 16Z_{1261} - 8Z_{1321} - 8Z_{1322} \\
&-14Z_{1331} - 14Z_{1332} - 7Z_{1341} - 7Z_{1342} - 19Z_{1351} - 19Z_{1352} - 16Z_{1361} - 16Z_{1362} - 8Z_{1421} - 8Z_{1422} \\
&-8Z_{1423} - 14Z_{1431} - 14Z_{1432} - 14Z_{1433} - 7Z_{1441} - 7Z_{1442} - 7Z_{1443} - 19Z_{1451} - 19Z_{1452} - 19Z_{1453} \\
&-16Z_{1461} - 16Z_{1462} - 16Z_{1463} + 8Z_{1525} + 8Z_{1526} + 14Z_{1535} + 14Z_{1536} + 7Z_{1545} + 7Z_{1546} + 19Z_{1555} \\
&+19Z_{1556} + 16Z_{1565} + 16Z_{1566} + 8Z_{1626} + 14Z_{1636} + 7Z_{1646} + 19Z_{1656} + 16Z_{1666} = 15
\end{aligned}$$

$$\begin{aligned}
&C_2^l - 51X_{25} - 51X_{26} - 12Z_{2211} - 9Z_{2231} - 5Z_{2241} - 15Z_{2251} - 10Z_{2261} - 12Z_{2311} - 12Z_{1312} \\
&-9Z_{2331} - 9Z_{2332} - 5Z_{2341} - 5Z_{2342} - 15Z_{2351} - 15Z_{2352} - 10Z_{2361} - 10Z_{2362} - 12Z_{2421} - 12Z_{2422} \\
&-12Z_{2423} - 9Z_{2431} - 9Z_{2432} - 9Z_{2433} - 5Z_{2441} - 5Z_{2442} - 5Z_{2443} - 15Z_{2451} - 15Z_{2452} - 15Z_{2453} \\
&-10Z_{2461} - 10Z_{2462} - 10Z_{2463} + 12Z_{2525} + 12Z_{2526} + 9Z_{2535} + 9Z_{2536} + 5Z_{2545} + 5Z_{2546} + 15Z_{2555} \\
&+15Z_{2556} + 10Z_{2565} + 10Z_{1566} + 12Z_{2626} + 12Z_{2636} + 5Z_{2646} + 15Z_{2656} + 10Z_{2666} = 7
\end{aligned}$$

$$\begin{aligned}
&C_2^u - 71X_{25} - 71X_{26} - 15Z_{2211} - 14Z_{2231} - 7Z_{2241} - 19Z_{2251} - 16Z_{2261} - 15Z_{2311} - 15Z_{1312} \\
&-14Z_{2331} - 14Z_{2332} - 7Z_{2341} - 7Z_{2342} - 19Z_{2351} - 19Z_{2352} - 16Z_{2361} - 16Z_{2362} - 15Z_{2421} - 15Z_{2422} \\
&-15Z_{2423} - 14Z_{2431} - 14Z_{2432} - 14Z_{2433} - 7Z_{2441} - 7Z_{2442} - 7Z_{2443} - 19Z_{2451} - 19Z_{2452} - 19Z_{2453} \\
&-16Z_{2461} - 16Z_{2462} - 16Z_{2463} + 15Z_{2525} + 15Z_{2526} + 14Z_{2535} + 14Z_{2536} + 7Z_{2545} + 7Z_{2546} + 19Z_{2555}
\end{aligned}$$

$$+19Z_{2556}+16Z_{2565}+16Z_{1566}+15Z_{2626}+15Z_{2636}+7Z_{2646}+19Z_{2656}+16Z_{2666}=8$$

$$C_3^l-49X_{35}-49X_{36}-12Z_{3211}-7Z_{3231}-5Z_{3241}-15Z_{3251}-10Z_{3261}-12Z_{3311}-12Z_{3312}$$
$$-7Z_{3321}-7Z_{3322}-5Z_{3341}-5Z_{3342}-15Z_{3351}-15Z_{3352}-10Z_{3361}-10Z_{3362}-12Z_{3411}-12Z_{3412}$$
$$-12Z_{3413}-7Z_{3421}-7Z_{3422}-7Z_{3423}-5Z_{3441}-5Z_{3442}-5Z_{3443}-15Z_{3451}-15Z_{3452}-15Z_{3453}$$
$$-10Z_{3461}-10Z_{3462}-10Z_{3463}+12Z_{3515}+12Z_{3516}+7Z_{3525}+7Z_{3526}+5Z_{3545}+5Z_{3546}+15Z_{3555}$$
$$+15Z_{3556}+10Z_{3565}+10Z_{3566}+12Z_{3616}+7Z_{3626}+5Z_{3646}+15Z_{3656}+10Z_{3666}=9$$

$$C_3^u-65X_{35}-65X_{36}-15Z_{3211}-8Z_{3231}-7Z_{3241}-19Z_{3251}-16Z_{3261}-15Z_{3311}-15Z_{3312}$$
$$-8Z_{3321}-8Z_{3322}-7Z_{3341}-7Z_{3342}-19Z_{3351}-19Z_{3352}-16Z_{3361}-16Z_{3362}-15Z_{3411}-15Z_{3412}$$
$$-15Z_{3413}-8Z_{3421}-8Z_{3422}-8Z_{3423}-7Z_{3441}-7Z_{3442}-7Z_{3443}-19Z_{3451}-19Z_{3452}-19Z_{3453}$$
$$-16Z_{3461}-16Z_{3462}-16Z_{3463}+15Z_{3515}+15Z_{3516}+8Z_{3525}+8Z_{3526}+7Z_{3545}+7Z_{3546}+19Z_{3555}$$
$$+19Z_{3556}+16Z_{3565}+16Z_{3566}+15Z_{3616}+8Z_{3626}+7Z_{3646}+19Z_{3656}+16Z_{3666}=14$$

$$C_4^l-53X_{45}-53X_{46}-12Z_{4211}-7Z_{4221}-9Z_{4231}-15Z_{4251}-10Z_{4261}-12Z_{4311}-12Z_{4312}$$
$$-7Z_{4321}-7Z_{4322}-9Z_{4331}-9Z_{4332}-15Z_{4351}-15Z_{4352}-10Z_{4361}-10Z_{4362}-12Z_{4411}-12Z_{4412}$$
$$-12Z_{4413}-7Z_{4421}-7Z_{4422}-7Z_{4423}-9Z_{4431}-9Z_{4432}-9Z_{4433}-15Z_{4451}-15Z_{4452}-15Z_{4453}$$
$$-10Z_{4461}-10Z_{4462}-10Z_{4463}+12Z_{4515}+12Z_{4516}+7Z_{4525}+7Z_{4526}+9Z_{4535}+9Z_{4536}+15Z_{4555}$$
$$+15Z_{4556}+10Z_{4565}+10Z_{4566}+12Z_{4616}+7Z_{4626}+9Z_{4636}+15Z_{4656}+10Z_{4666}=5$$

$$C_4^u-72X_{45}-72X_{46}-15Z_{4211}-8Z_{4221}-14Z_{4231}-19Z_{4251}-16Z_{4261}-15Z_{4311}-15Z_{4312}$$
$$-8Z_{4321}-8Z_{4322}-14Z_{4331}-14Z_{4332}-19Z_{4351}-19Z_{4352}-16Z_{4361}-16Z_{4362}-15Z_{4411}-15Z_{4412}$$
$$-15Z_{4413}-8Z_{4421}-8Z_{4422}-8Z_{4423}-14Z_{4431}-14Z_{4432}-14Z_{4433}-19Z_{4451}-19Z_{4452}-19Z_{4453}$$
$$-16Z_{4461}-16Z_{4462}-16Z_{4463}+15Z_{4515}+15Z_{4516}+8Z_{4525}+8Z_{4526}+14Z_{4535}+14Z_{4536}+19Z_{4555}$$
$$+19Z_{4556}+16Z_{4565}+16Z_{4566}+15Z_{4616}+8Z_{4626}+14Z_{4636}+19Z_{4656}+16Z_{4666}=7$$

$$C_5^l-43X_{55}-43X_{56}-12Z_{5211}-7Z_{5221}-9Z_{5231}-5Z_{5241}-10Z_{5261}-12Z_{5311}-12Z_{5312}$$

$$-7Z_{5321}-7Z_{5322}-9Z_{5331}-9Z_{5332}-5Z_{5341}-5Z_{5342}-10Z_{5361}-10Z_{5362}-12Z_{5411}-12Z_{5412}$$
$$-12Z_{5413}-7Z_{5421}-7Z_{5422}-7Z_{5423}-9Z_{5431}-9Z_{5432}-9Z_{5433}-5Z_{5441}-5Z_{5442}-5Z_{5443}$$
$$-10Z_{5461}-10Z_{5462}-10Z_{5463}+12Z_{5515}+12Z_{5516}+7Z_{5525}+7Z_{5526}+9Z_{5535}+9Z_{5536}+5Z_{5545}$$
$$+5Z_{5546}+10Z_{5565}+10Z_{5566}+12Z_{5616}+7Z_{5626}+9Z_{5636}+5Z_{5646}+10Z_{5666}=15$$

$$C_5^u-60X_{55}-60X_{56}-15Z_{5211}-8Z_{5221}-14Z_{5231}-7Z_{5241}-16Z_{5261}-15Z_{5311}-15Z_{5312}$$
$$-8Z_{5321}-8Z_{5322}-14Z_{5331}-14Z_{5332}-7Z_{5341}-7Z_{5342}-16Z_{5361}-16Z_{5362}-15Z_{5411}-15Z_{5412}$$
$$-15Z_{5413}-8Z_{5421}-8Z_{5422}-8Z_{5423}-14Z_{5431}-14Z_{5432}-14Z_{5433}-7Z_{5441}-7Z_{5442}-7Z_{5443}$$
$$-16Z_{5461}-16Z_{5462}-16Z_{5463}+15Z_{5515}+15Z_{5516}+8Z_{5525}+8Z_{5526}+14Z_{5535}+14Z_{5536}+7Z_{5545}$$
$$+7Z_{5546}+16Z_{5565}+16Z_{5566}+15Z_{5616}+8Z_{5626}+14Z_{5636}+7Z_{5646}+16Z_{5666}=19$$

$$C_6^l-48X_{65}-48X_{66}-12Z_{6211}-7Z_{6221}-9Z_{6231}-5Z_{6241}-15Z_{6251}-12Z_{6311}-12Z_{6312}$$
$$-7Z_{6321}-7Z_{6322}-9Z_{6331}-9Z_{6332}-5Z_{6341}-5Z_{6342}-15Z_{6351}-15Z_{6352}-12Z_{6411}-12Z_{6412}$$
$$-12Z_{6413}-7Z_{6421}-7Z_{6422}-7Z_{6423}-9Z_{6431}-9Z_{6432}-9Z_{6433}-5Z_{6441}-5Z_{6442}-5Z_{6443}$$
$$-15Z_{6451}-15Z_{6452}-15Z_{6453}+12Z_{6515}+12Z_{6516}+7Z_{6525}+7Z_{6526}+9Z_{6535}+9Z_{6536}+5Z_{6545}$$
$$+5Z_{6546}+15Z_{6555}+15Z_{6555}+12Z_{6616}+7Z_{6626}+9Z_{6636}+5Z_{6646}+15Z_{6656}=10$$

$$C_6^u-63X_{65}-63X_{66}-15Z_{6211}-8Z_{6221}-14Z_{6231}-14Z_{6241}-19Z_{6251}-15Z_{6311}-15Z_{6312}$$
$$-8Z_{6321}-8Z_{6322}-14Z_{6331}-14Z_{6332}-7Z_{6341}-7Z_{6342}-19Z_{6351}-19Z_{6352}-15Z_{6411}-15Z_{6412}$$
$$-15Z_{6413}-8Z_{6421}-8Z_{6422}-8Z_{6423}-14Z_{6431}-14Z_{6432}-14Z_{6433}-7Z_{6441}-7Z_{6442}-7Z_{6443}$$
$$-19Z_{6451}-19Z_{6452}-19Z_{6453}+15Z_{6515}+15Z_{6516}+8Z_{6525}+8Z_{6526}+14Z_{6535}+14Z_{6536}+7Z_{6545}$$
$$+7Z_{6546}+19Z_{6555}+19Z_{6555}+15Z_{6616}+8Z_{6626}+14Z_{6636}+7Z_{6646}+19Z_{6656}=16$$

$$Z_{ijkl} \le X_{ij} \qquad (a)$$
$$Z_{ijkl} \le X_{kl} \qquad (b)$$
$$X_{ij}+X_{kl}-1 \le Z_{ijkl} \qquad (c)$$

$$C_1^l-100TL_1 \le 17$$

$$C_{2^l} - 100TL_2 \leq 8$$
$$C_3^l - 100TL_3 \leq 15$$
$$C_{4^l} - 100TL_4 \leq 7$$
$$C_5^l - 100TL_5 \leq 25$$
$$C_6^l - 100TL_6 \leq 32$$

$$100TL_1 - C_1^l \leq 82.99$$
$$100TL_2 - C_2^l \leq 91.99$$
$$100TL_3 - C_3^l \leq 84.99$$
$$100TL_4 - C_4^l \leq 92.99$$
$$100TL_5 - C_5^l \leq 74.99$$
$$100TL_6 - C_6^l \leq 67.99$$

$$100E_1 + C_1^u \geq 17.01$$
$$100E_2 + C_2^u \geq 8.01$$
$$100E_3 + C_3^u \geq 15.01$$
$$100E_4 + C_4^u \geq 7.01$$
$$100E_5 + C_5^u \geq 25.01$$
$$100E_6 + C_6^u \geq 32.01$$

$$100E_1 + C_1^u \leq 117$$
$$100E_2 + C_2^u \leq 108$$
$$100E_3 + C_3^u \leq 115$$
$$100E_4 + C_4^u \leq 107$$
$$100E_5 + C_5^u \leq 125$$
$$100E_6 + C_6^u \leq 132$$

$$TL_1 + PT_1 + E_1 = 1$$
$$TL_2 + PT_2 + E_2 = 1$$
$$TL_3 + PT_3 + E_3 = 1$$

$$TL_4 + PT_4 + E_4 = 1$$
$$TL_5 + PT_5 + E_5 = 1$$
$$TL_6 + PT_6 + E_6 = 1$$

$$X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} = 1$$
$$X_{21} + X_{22} + X_{23} + X_{24} + X_{25} + X_{26} = 1$$
$$X_{31} + X_{32} + X_{33} + X_{34} + X_{35} + X_{36} = 1$$
$$X_{41} + X_{42} + X_{43} + X_{44} + X_{45} + X_{46} = 1$$
$$X_{51} + X_{52} + X_{53} + X_{54} + X_{55} + X_{56} = 1$$
$$X_{61} + X_{62} + X_{63} + X_{64} + X_{65} + X_{66} = 1$$
$$X_{11} + X_{21} + X_{31} + X_{41} + X_{51} + X_{61} = 1$$
$$X_{12} + X_{22} + X_{32} + X_{42} + X_{52} + X_{62} = 1$$
$$X_{13} + X_{23} + X_{33} + X_{43} + X_{53} + X_{63} = 1$$
$$X_{14} + X_{24} + X_{34} + X_{44} + X_{54} + X_{64} = 1$$
$$X_{15} + X_{25} + X_{35} + X_{45} + X_{55} + X_{65} = 1$$
$$X_{16} + X_{26} + X_{36} + X_{46} + X_{56} + X_{66} = 1$$
$$X_{ij}, E_i, TL_i, PT_i = 0 \text{ or } 1$$

Bibliography

[1] Andreas Drexyl , " Scheduling of project networks by job assignment". *Management Science*, 37(12):1590-1602, 1991.

[2] Bintong Chen, Mustafa, C. Pinar, Barbaros Tansel and Hamde Yaman "Project management with interval activity durations", *Turkey Conference Paper*, 24-26, 1998.

[3] Bagchi, U., Sullivan, R. S., and Chang, Y. L., "Minimizing mean absolute deviation of completion times about a common due date". *Naval Research Logistics Quarterly*, 33:227-240, 1986.

[4] Burt, J.M., and Garman, M.B. " Conditional Monte Carlo, a simulation technique for stochastic network analysis". *Management Science*, 18:207-217, 1971.

[5] Clark C.E., "The greatest of a finite set of random variables". *Operations Research*, 9:145-162, 1981.

[6] Cleef, H.j., and Gaul, W., "Project Scheduling by stochastic programming". *Mathematische Operations forscchung and Statistik, Series Optimization*, 13:449-456, 1982.

[7] Conway, R. W., Maxwell, W. L. and Miller E.W., "Theory of Scheduling". *Addison-Wesley, Reading, Mass*, p 26, 1967.

[8] Cynthia McCahon., "Using PERT as an approximation of fuzzy project-network analysis", *IEEE Transactions on Engineering Management*, 40(2):146-153, 1993.

[9] Dubois, D., and Prade, H. "Operations on fuzzy numbers", *International Journal for Systems Science*, 9:613-626, 1978a.

[10] Dubois, D., and Prade, H. "Algorithmes de plus courts chemins pour traiter des donnees floues", *RAIRO/Rescherche Operationelle*, 2:213-227, 1978b.

[11] Elmaghraby, S.E., "On the expected durations of PERT type networks". *Management Sceince*, 5:299-306, 1967.

[12] Elmaghraby, S.E., "Activity Networks". *Wiley, New York*, 1977.

[13] Elsayed A. Elsayed and Thomas O. Boucher, "Analysis and Control of Production Systems". *Prentice Hall Inc., Eagle wood Cliffs, N.J. 07632*, 1982.

[14] Euler L., " The Konigsberg Bridges". *Scientific America*, 189, pp 66-70 , 1953.

[15] Fulkerson, D.R., "Expected critical path in PERT type networks", *Operations Research*10:808-817, 1962.

[16] Gaul, W., "Bounds for the expected durations of a stochastic project planning model", *Journal of Information and Optimization Sciences*, 2:45-63, 1981.

[17] Glover, Fred, and E. Woolsey, " Converting 0-1 polynomial programming problem to a 0-1 linear program". *Operations Research*, 22:180-1182, 1974.

[18] Hamdy A. Taha, " Operations Research An Introduction". *Macmillan Publishing company, 866 Third Avenue, NY 10022*, 1991.

[19] Hardy, Littlewood, and Polya., "Inequalities", *Second Edition, New York: Cambridge University Press*, page 261, 1952.

[20] Hartley H.O., and Worthman, A.W., "A statistical theory for PERT critical path analysis", *Management Science*, 12:469-481, 1966.

[21] Heinrich, J. Rommelfanger, "Network analysis and information flow in fuzzy environment", *Fuzzy Sets and Systems*, 67:119-128, 1994.

[22] Hitchcock F.L., " The Distribution of a Product from Several Sources to Numerous Localities". *Journal of Mathematics and Physics.*, 20, pp 224-230, 1941.

[23] Jackson, J.R. , " Scheduling a Production Line to Minimize Maximum Tardiness". *Research Report*, University of California at Los Angeles, 1983.

[24] Jeff L. Kennington and Richard V. Helgason, " Algorithms for Network Programming". *John Wiley & Sons.*, 1980.

[25] Kanet, J. K., "Minimizing the average deviation of job completion times about a common due date". *Naval Research Logistics Quarterly*, 28:643-651, 1981.

[26] Kelly J.E.Jr. and M.R. Walker, "Critical Path Planning and Scheduling". *Proceeding of Eastern Joint Computer Conference, Boston*, 1959.

[27] Kenneth, R. Baker , " Introduction to Sequencing and Scheduling". *John Wiley & Sons*, 1983.

[28] Ki Hang Kim and Fred W. Roush., "Fuzzy flows on Networks". *Fuzzy Sets and Systems*, 8:35-38, 1982.

[29] Kise, H., Ibraki, T. and Mine, H. , "A Solvable Case of the One Machine Scheduling Scheduling Problems with Ready Times and Due Dates". *Operations Research*, Vol-26:121-126, 1978.

[30] Kleindorfer, G.B., "Bounding distributions for a stochastic, acyclic nework", *Operations Research*, 19:1586-1601, 1971.

[31] Klein Haneveld, W.K., "Robustness against dependence in PERT: An application of duality and distributions with known marginals", *Mathematical Programming Study*, 27:153-182, 1986.

[32] Koopmans T.C., "Optimum Utilization of the Transportation Systems". *Proceedings of International Statistical Conference, Washington D.C.*, 1947.

[33] Kotiah, T.C.T. and N.D. Wallace, "Another look at PERT assumptions", *Management Science*, 20(1):443-449, 1973.

[34] Lootsma, F.A., "Network planning with stochastic activity durations, an evaluation of PERT", *Statistica Neerlandica*, 20:43-69, 1966.

[35] Malcom, D.G., Roseboom, J.H., Clark, C.E., and Fazar, W., "Application for a technique for research and development proram evaluation", *Operations Research*, 7:646-669, 1959.

[36] MacCrimmon, K.R., and Ryavec, C.A., "An analytical study of PERT assumptions", *Operations Research*, 12:16-37, 1964.

[37] Martin, J.J., "Distribution of the time through a directed, acyclic network". *Operations Research*, 13:46-66, 1965.

[38] Meilijson, I., and Nadas, A., "Convex majorization with an application to the length of critical paths", *Journal of Applied Probability*, 16:671-677, 1979.

[39] Mokhtar S. Bazaraa, John J. Jarvis and Hanif D. Sherali, " Linear Programmming and Network Flows". *John Wiley & Sons*, 1990.

[40] Moore J. M., "An n-job , one-machine Sequencing Algorithm for minimizing number of late jobs". *Management Science*, 15:102-109, 1968.

[41] Nadas, A., "Probablistic PERT", *IBM Journal of Research and Development*, 23:239-258, 1979.

[42] Paul A. Jenson and J. Wesley Barnes, "Network Flow Programming". *Robert E. Kreigner Publishing Company, Inc.*,, 1987.

[43] Parks, W.H., and Ramsing, K.D., "The use of compound Poisson in PERT", *Management Science*, 15(8):B-397-B402, 1969.

[44] "PERT Summary Report", *Bureau of Ordinance, Department of the Navy, Washington, DC.* , 1958.

[45] Don T. Phillips, Raghavendran A. and James J. Solberg, "Operations Research: Principles and Practice". *John Wiley & Sons*, 1976.

[46] Rinnooy Kan A. H. G., "Machine Scheduling Problems: Classification, Complexity and Computations", *us Nijhoff, The Hague, Holland*, p 68,1976.

[47] Robillard, P., and Traham. M., "The completion times of PERT networks", *Operations Research*, 25:15-29, 1977.

[48] Shogan, A.W., "Bounding distributions for a stochastic PERT network", *Networks*, 7:359-381, 1977.

[49] Simon French, " Sequencing and Scheduling : An Introduction to the Mathematics of the Job-Shop". *John Wiley & Sons, New York*, 1986.

[50] Sofjan H. Nasution., "Fuzzy critical path method". *IEEE Transactions on Systems, Man, and Cybernetics*, 24(1):48-57, 1994.

[51] Sturm, L.J.M., " A simple Optimality Proof of Moore's Algorithm". *Management Science*, Vol-17(1), March, 1956.

[52] Sunderaraghavan, P., and M. Ahmed., "Minimizing the sum of absolute lateness in single machine and multimachine scheduling". *Naval Research Logistics Quarterly*, 31:325-333, 1984.

[53] Zadeh, L.A., "Fuzzy sets", *Information and Control*, 8:338-353, 1965.

Vita

- Shaikh Arifusalam
- Born in Hyderabad, Andhra Pradesh, India on August 28, 1974.
- Received Bachelor's degree in Mechanical Engineering with specialization in Production Engineering, from Osmania University, Hyderabad, India in June, 1994.
- Completed Master's degree requirements in Systems Engineering Department at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in December, 1998.
- Research areas include Network Modeling, Scheduling and Operations Research.